COLOSSUS
COLLABORATIVE SYSTEM OF SYSTEMS

DELIVERABLE D 5.1

SOTA Holistic SoS Engineering

Author(s): Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
WorkPackage N°: WP5.2
Due date of deliverable: 31/10/2023
Actual submission date: 31/01/2024
Document ID: COLOSSUS_D5-1_SOTA Holistic SoS Engineering_v1.2.docx

Grant Agreement number: 101097120
Project acronym: COLOSSUS
Project title: Collaborative System of Systems Exploration of Aviation Products, Services and Business Models
Start date of the project: 01/02/2023
Duration: 36 months
Project coordinator name & organisation:
Prajwal Shiva Prakasha
DLR - German Aerospace Center Institute of System Architectures in Aeronautics | Aviation System Design and Assessment
Tel: +49 40 2489641-322
E-mail: prajwal.prakasha@dlr.de

FUNDED BY THE EUROPEAN UNION UNDER GRANT AGREEMENT NO 101097120. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR CINEA. NEITHER THE EUROPEAN UNION NOR THE GRANTING AUTHORITY CAN BE HELD RESPONSIBLE FOR THEM.

Funded by the European Union

Project website address: https://colossus-sos-project.eu/

DOCUMENT INFORMATION

Document ID	COLOSSUS_D5-1_SOTA Holistic SoS Engineering_v1.2.docx
Version	1.2
Version Date	31/01/2024
Author	Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
Dissemination level	PU

APPROVALS

	Name	Company	Date
Coordinator	Prajwal Shiva Prakasha	German Aerospace Center (DLR)	31/01/2024
WP Leader	Ludvig Knöös Franzén	Linköping University (LiU)	31/01/2024

REVIEWERS

Name	Company	Date
Chengxi Liu, Magnus Eek	The Swedish National Road and Transport Research Institute (VTI)	23/01/2024
Marta Ribeiro	Delft University of Technology (TUD)	23/01/2024
Nabih Naeem	German Aerospace Center (DLR)	31/01/2024

DOCUMENTS HISTORY

Version	Date	Modification	Authors
0.1	17/08/2023	Initialized document	Ludvig Knöös Franzén
0.2	12/12/2023	Added two-part literature study	Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
0.3	13/01/2024	Finalized document for first review	Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
1.0	15/01/2024	Version sent for first review	Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
1.1	31/01/2024	Version with review comments incorporated	Ludvig Knöös Franzén, Jorge Lovaco, Felipe Villas
1.2	31/01/2024	Final comments incorporated; version sent for delivery	Ludvig Knöös Franzén

LIST OF AUTHORS

Full Name	Organisation
Ludvig Knöös Franzén	Linköping University
Jorge Lovaco	Linköping University
Felipe Villas	Linköping University

DISTRIBUTION LIST

Full Name	Organisation
Magnus Eek	VTI
Chengxi Liu	VTI
Marta Ribeiro	TUDelft
Nabih Naeem	DLR
Prajwal Shiva Prakasha	DLR
COLOSSUS Consortium	-

TABLE OF CONTENTS

COLOSSUS
COLLABORATIVE SYSTEM OF SYSTEMS

GLOSSARY

Acronym	Signification
ABM	Agent-Based Model
ABMS	Agent-Based Modelling & Simulation
ABS	Agent-Based Simulation
AI	Artificial Intelligence
COLOSSUS	Collaborative System of Systems Exploration of Aviation Products, Services & Business Models
CONOPS	Concept Of Operations
CS	Constituent System
DDNA	Development Dependency Network Analysis
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
DoE	Design of Experiments
FDNA	Functional Dependency Network Analysis
GIS	Geographic Information System
MBSE	Model-Based Systems Engineering
MDO	Multidisciplinary Design Optimization
ME	Mission Engineering
ML	Machine Learning
MoDAF	Ministry of Defence Architecture Framework
SE	Systems Engineering
SoS	System-of-Systems
SoSE	System-of-Systems Engineering
SOTA	State-Of-The-Art
SysML	Systems Modelling Language
UAF	Unified Architecture Framework

ABSTRACT

Deliverable 5.1 provides a State-Of-The-Art (SOTA) on holistic System-of-Systems Engineering (SoSE) within Task 5.2 for the COLOSSUS project. The foundation for the SOTA is a questionnaire distributed to the COLOSSUS partners with prior knowledge of SoSE Engineering methodologies and agent-based modelling and simulation techniques. Additionally, a literature study on the same topics and methodologies, but outside the COLOSSUS consortium, acts as a complementary survey to the current SOTA overview. Together, the SOTA answers from the distributed questionnaire and the results of the literature study serve as a base for an overarching frame of reference for the COLOSSUS project in this deliverable.

1. EXECUTIVE SUMMARY

1.1 Introduction

System-of-Systems (SoS) is a concept that is gaining importance in diverse engineering domains. An SoS refers to a collection of individual systems, referred to as constituent systems (CS), that are interconnected and working together to achieve a common goal. In an SoS, each CS maintains its independence and has its own purpose and functionality, but when integrated into the larger system, it may collaborate with other CSs to provide additional capabilities, or address complex challenges, that cannot be efficiently handled by any of the individual systems alone.

The discipline for integrating systems into an SoS is referred to as System-of-Systems Engineering (SoSE). SoSE can thereby be seen as a methodology that focuses on the interplay between CS in order to achieve unique capabilities through collaborative efforts. However, SoSE implies a more holistic approach where additional aspects, such as the influence of the operational environment, must be considered. It also requires a proper understanding of the problem at hand and all the involved layers of an SoS that are influenced. SoSE is therefore a means by which to derive the architecture for such SoS solutions.

This deliverable document presents a State-Of-The-Art (SOTA) investigation of different methodologies and tools that may enable a holistic system-of-systems engineering methodology where a multitude of aspects also are considered. The aim of this document is two-fold:

- To present a SOTA for holistic SoSE. This will be based upon the combined experience of relevant COLOSSUS partners, as well as on a literature study on the current SOTA of the field in general.
- To suggest a first version of a frame of reference which may act as a foundation for decision making of the overall SoS architecting approach and implementation to-be-realized within the COLOSSUS project.

As a result, this document provides both an overview of the current research forefront and industrial SOTAs, and a frame of reference for SoSE methodologies that may support the holistic SoS architecting processes to be performed within COLOSSUS.

1.2 Brief description of the work performed and results achieved

The performed work involved the distribution of a questionnaire which was designed to capture the knowledge and current research forefront in SoSE from all participating COLOSSUS partners. The distributed questionnaire was also complemented with a literature study of other initiatives in order to identify the current research state of SoSE and the existing research gaps. The answers to the distributed questionnaire and the performed literature study together illustrate the current capabilities in the research field. Consequently, this deliverable contributes a frame of reference upon which the COLOSSUS project can build and cover some of the gaps in current holistic SoSE practices.

1.3 Deviation from the original objectives

1.3.1 Description of the deviation

Not all COLOSSUS partners have previous experience in SoSE or agent-based modelling/simulation (ABM/ABS) and have therefore been unable to provide answers to some of the questions in the supplied questionnaire.

This deliverable document has also seen a delay and deviation from its original due date. This has partly been caused by less manpower than expected at the beginning of the project.

1.3.2 Corrective actions

To get a broader view on the current SOTA of the field, the questionnaire answers were complemented with a literature study on the SOTA for SoSE and ABM/ABS. This was also done in order to get the additional perspective of relevant actors not included in the COLOSSUS project.

Corrective actions for the deliverable delay have been performed by providing the questionnaire answers and continuously updating found references on our shared online workspace. This has been done so that the COLOSSUS consortium in general always could access the “raw” material that this deliverable is based upon.

2. WORK PERFORMED

The work that was performed for this deliverable can be split into two parts; a collection of SOTA for SoSE from COLOSSUS partners through a distributed questionnaire, and a collection of SOTA for SoSE through a short literature study (to capture any advancements in the field by non-COLOSSUS related organizations and actors). This section presents the SOTA collection work that was performed for the deliverable, starting with the layout of the distributed questionnaire.

2.1 Layout of the distributed questionnaire

The questionnaire, which supported the SOTA collection from involved COLOSSUS partners, was divided into two main categories, namely:

- Methodologies for holistic SoS engineering
- Experience in agent-based modelling and simulation

The questionnaire itself consisted of the following questions for both categories mentioned above:

1. **General Information**
    a. Organization/Company
    b. Edited by
2. **System-of-Systems Engineering Methodologies**
    a. What methodologies do you use in System of Systems Engineering (SoSE)?
    b. What tools do you use in SoSE?
    c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?
    d. What knowledge do you have of current state-of-the-art (SOTA) advantages or disadvantages of different methodologies and their applicability to your organization/company?
    e. Are there some best practices in SoSE that you know of? If yes, which?
    f. Which of your (if any) listed methodologies do you think are of most relevance within the COLOSSUS project and why?
    g. Are there any SoSE methodologies, or similar, that you are aware of but would like to investigate in more detail under the COLOSSUS project?
3. **Agent-Based Modelling and Simulation Experience**
    a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?
    b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?
    c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?
    d. Have you integrated other modelling techniques (e.g., machine learning, network analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.
    e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.
4. **Miscellaneous**
    a. Do you have any suggestions/recommendations on relevant literature for state-of-the-art (SOTA) in connection to SoSE?
    b. Any additional comments/thoughts?

2.2 Complementary literature study

In addition to the distributed questionnaire, a literature study was performed to gain a broader view of SOTA holistic SoS Engineering from actors outside the COLOSSUS project. Similar to the questionnaire, the literature study was split into the same parts as in the previous section:

- Literature study on SOTA for methodologies on SoS analyses and SoS engineering
- Literature study on SOTA for agent-based modelling and simulation techniques

The aim of the literature studies was to enhance understanding of the current knowledge landscape outside the COLOSSUS project. The goal was to identify potential gaps or areas for further investigation and highlight additional approaches and methods for realizing holistic analyses. Both literature studies were performed based on the following steps:

1. **Survey of existing literature:**
   Collecting relevant literature from sources such as journal articles, conference contributions, books and more in relation to the research topic in question.

2. **Analysis and Synthesis:**
   Identification of common denominators between works, for example findings, patterns and methodologies.

3. **Identification of Gaps and Trends**
   Finding and identifying, for example, questions that have not yet been answered, as well as conflicting information in order to position the work presented in this deliverable in the broader context.

4. **Evaluation and Organization**
   To ensure that the collected knowledge is reliable and of good quality before being organized for the subsequent contextualization.

5. **Contextualization**
   The final step puts the findings of the literature study into the context of the current knowledge and understanding of the topic.

The collected answers from the questionnaires could thereafter be complemented with the literature study results and finally synthesized into this document and frame of reference for the envisioned holistic SoSE methodology to be developed.

3. Questionnaire results from each participating partner

The answers to the questionnaire outlined in section 2.1 by participating partners of COLOSSUS are presented in this section. Due to the varying degrees of prior SoSE and ABM/ABS knowledge from each partner, only the questions answered are presented. Additionally, some answers have seen minor adjustments for improved readability in this deliverable document.

3.1 CFSE

1. **General Information:**

Question	Answer
Organization/Company	*CFS Engineering*
Edited by	*Jan Vos and Giacomo Benedetti*

2. **System-of-Systems Engineering Methodologies:**

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*Not applicable.*
2b. What tools do you use in SoSE?	*Not applicable.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	*Automatic CAD repair and mesh generation.*
2d. What knowledge do you have of current state-of-the-art (SOTA), advantages, disadvantages of different methodologies and their applicability to your organization/company?	*Pretty good knowledge about the SOTA of aerodynamics simulation practices.*

3. **Agent-Based Modelling and Simulation Experience:** *Not applicable.*

4. **Miscellaneous:**

Question	Answer
4b. Any additional comments/thoughts?	*Translating the general ABS terminology to the SoS application cases will give a better understanding of it.*

3.2 DLR

1. General Information:

Question	Answer
Organization/Company	*German Aerospace Center (DLR)*
Edited by	*Jasamin Akbari, Nazlican Cigal and Nabih Naeem*

2. System-of-Systems Engineering Methodologies:

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*The Systems Modelling Language (SysML) as a modelling language was used based on the Object-Oriented Systems Engineering Method (OOSEM). The SysML profile is limited in some of its notations (e.g., no stakeholders). Some modifications were made to meet the requirements for an SoS methodology. On the one hand, the operational analysis aspect was added (ConOps). On the other hand, the dimensions of an SoS (constituent system, subsystem) were distinguished. SysML v2 is also utilized.* *The "dual V" and CUBE methodologies were considered in the research activities.* *For the functional and logical architecting, the function induction concept was explored.* *An ontology can be used to describe the high-level interactions between different system elements or entities (behaviour, ConOps, SoS, subsystem and functions).*
2b. What tools do you use in SoSE?	*Cameo Systems Modeler with SysML v1.6. Also, SysML v2.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	*One main aspect of systems engineering generally, but also in SoSE, is the identification of functions and capabilities, as well as their decomposition or the induction of new functions.* *Also, the behavioural perspective to evaluate the developed architecture is missing (to be fulfilled with ABS, hopefully).*
2d. What knowledge do you have of current state-of-the-art (SOTA), advantages, disadvantages of different methodologies and their applicability to your organization/company?	*We gained an overview of different frameworks which can describe the operational and high-level aspects:* • *Unified Architecture Framework (UAF)* • *Ministry of Defence Architecture Framework (MoDAF)*

	• *Department of Defense Architecture Framework (DoDAF)* *Although these frameworks are referred to as architectural frameworks, we are missing the architecting activities. However, they can deliver nice viewpoints, especially when describing a large SoS with many domains involved. Unfortunately, we did not have access to the UAF profile within Cameo to test it more.*
2e. Are there some best practices in SoSE that you know of? If yes, which?	*Network-based engineering (as in ontologies).*
2f. Which of your (if any) listed methodologies do you think are of most relevance within the COLOSSUS project and why?	*For the high-level description of the system, SysML with an underlying SoS methodology can provide an initial overview of the SoS (SoS identification).* *Functional architecting and considering capabilities as overarching functionalities of the SoS should be part of the COLOSSUS project. Different concepts (F/M tree, function induction) should be considered and compared.*
2g. Are there any SoSE methodologies, or similar, that you are aware of but would like to investigate in more detail under the COLOSSUS project?	*The Architectural Design Graph (ASDG) was developed during the AGILE project. It enables an architectural design space modelling. For the COLOSSUS project we would like to expand the methodology to SoS and apply this methodology to an application case and explore the benefits and limitations.*

3. **Agent-Based Modelling and Simulation Experience:**

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*Development of ABS for Wildfire and UAM use cases for the purpose of holistic evaluation of SoS. ABS Framework advancement, agent logic modelling, execution of ABS studies using DoEs, results evaluation of SoS results.*
3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	*SoSID Toolkit, internally developed code base which is to be open sourced.* ***Strengths:*** • *Written in Python: Accessible for most engineers.* • *GUI available for debugging.* • *Headless execution of DoEs, performant result analysis.* • *Interactive debugging.* • *Integration with other tools is enabled and tested.* ***Weaknesses:*** • *Written in Python: Not the most performant language.* • *Difficult in testing ABMs in general, also applicable to SoSID toolkit.*

	• *May have a higher learning curve as it accounts for large-scale and complex systems.*
3c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?	*Verifying ABM behaviour can be a challenge especially with large and complex ABM. Simulating large-scale models may lead to high computational demands.*
3d. Have you integrated other modelling techniques (e.g., machine learning, network analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.	*No, we have not integrated the abovementioned techniques yet.*
3e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.	• *Follow SOLID coding principles.* • *Test Driven Development.* • *Object Oriented Programming.* • *Collaborative Development.*

4. **Miscellaneous:**

Question	Answer
4b. Any additional comments/thoughts?	*Different ranges of SoS (technical, socio-technical, enterprise) should be considered or at least it should be clarified for which range/type of system we are developing a framework/methodology.*

3.3 INCAS

1. **General Information:**

Question	Answer
Organization/Company	*Institutul Național de Cercetare-Dezvoltare Aerospațială "Elie Carafoli" (INCAS)*
Edited by	*Răzvan-Ionuț Bălașa, Marian-Ciprian Bîlu*

2. **System-of-Systems Engineering Methodologies:**

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*Our experience in SoS is connected to RoFSim (Dynamic Simulation System for Pilot-in-the-loop Applications), a robotic flight simulator.* *RoFSim has the following methodologies:* ***Scenario Generation:*** *RoFSim generates scenarios and environments that are used in training and test cases. It can generate*

	environmental conditions like weather, terrain features, and specific mission scenarios such as the loss of an engine or all engines, or a fire. ***Sensor Simulation:*** *RoFSim incorporates sensor simulation to replicate the input received from various sensors used in real aircraft. It simulates sensors such as GPS (Global Positioning System), altimeters, accelerometers, gyroscopes, magnetometers and other perception sensors that are used in navigation.* ***Control Algorithm Development:*** *RoFSim implements different control strategies, such as attitude simulation and acceleration simulation which uses a washout filter for control.* ***Data Logging and Analysis:*** *RoFSim not only records data from the simulated aircraft but also incorporates a human factors system that measures and records data on human performance and behaviour. This data analysis allows us to do performance evaluation, debugging and optimizations to the RoFSim.* ***Virtual Reality:*** *RoFSim utilizes virtual reality technology to create an immersive and realistic environment around the pilot. This technology also provides the added benefit of being lightweight. It enhances the sense of presence and enables the pilot to have a more intuitive interaction with the environment.* ***Physics-Based Simulation:*** *RoFSim employs physics-based simulation principles to model the behaviour of the aircraft and its interaction with the environment. These simulations use principles of aerodynamics, mechanics, and control theory to simulate the flight dynamics, forces, and interactions that affect the RoFSim's movement.*
2b. What tools do you use in SoSE?	*We use software created by us called RoFSim Controller that connects all the systems that are responsible for the movement of the simulator. It takes the flight data from the simulated aircraft and processes it using our control algorithm and the data from our safety system. After processing, it converts the data into a format that can be processed by the controller of the robotic arm. The software is developed in C#, and the robot utilizes a programming language called "Rapid".*

	RoFSim Controller also monitors the performance of the robotic arm and gives its status and operational parameters through a visualization capability.

3. Agent-Based Modelling and Simulation Experience:

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*INCAS has expertise in agent-based modelling and simulation, particularly in the realm of using Machine Learning Agents. We have extensive experience in training and deploying intelligent agents using machine learning algorithms. This enables us to create simulations that can learn and adapt based on their interactions with the environment. These simulations provide valuable insights, optimize processes, and support informed decision-making in various domains, including urban planning, transportation, and more. Our team is skilled in designing and implementing agent-based models, training intelligent agents, and analysing simulation results. We are equipped to tackle complex problems, tailor simulations to specific requirements, and deliver valuable solutions that leverage the potential of agent-based modelling and simulation.*
3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	*We have used a toolkit called Unity ML agents and implemented it in the Unity 3D Engine.* ***Strengths:*** • ***Machine Learning Capabilities****: Unity ML Agents provides built-in support for training agents using various machine learning algorithms, including reinforcement learning.* • ***Community and Resources****: Unity ML Agents has a big community of developers and researchers. This community shares resources and tutorials, making it easier to learn and leverage the tool effectively. Additionally, the availability of documentation helps overcome challenges and find solutions.* • ***Integration with Unity****: Unity ML Agents integrates with the Unity 3D engine, which offers a wide range of features and tools for creating realistic and immersive environments. This integration allows for visually appealing simulations with rich graphics and realistic physics simulations.* • ***Extensibility****: Unity ML Agents is a flexible and extensible framework, allowing developers to*

	customize its functionalities to suit specific simulation requirements. Developers can create complex agent behaviours and define rewards and punishments to enhance the realism and effectiveness of the simulations. ***Weaknesses:*** • ***Computational Requirements**: Training complex agent models using machine learning algorithms can be computationally intensive, requiring substantial computational resources, including GPUs.* • ***Learning Curve**: While Unity ML Agents has powerful capabilities, it can have a steep learning curve, especially for those new to machine learning. Understanding the underlying concepts, configuring simulations, and optimizing agent training can require a significant investment of time and effort.* • ***Documentation and Support**: Although Unity ML Agents benefit from a growing community and available resources, the documentation and support may not cover all use cases comprehensively. Exploring and adapting to new updates and features can present challenges without comprehensive documentation or dedicated support.* • ***Lack of Advanced Modelling Features**: Unity ML Agents primarily focuses on agent-based modelling and machine learning aspects. While it provides a solid foundation for simulating agent behaviours, it may lack some advanced modelling features found in specialized simulation tools. Incorporating domain-specific features may require additional customization or integration with external tools.*
3c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?	*Here are some challenges we have encountered:* ***Computational resources**: Training and running simulations can place significant computational demands, especially for large environments or complex agent behaviours, necessitating powerful hardware.* ***Performance constraints and scalability**: Scaling simulations to accommodate a larger number of agents or larger environments can present performance challenges, requiring efficient execution.*

	***Model complexity**: Designing agent-based models requires careful coding of agents' behaviours and interactions, which can be complex and challenging.* ***Sensitivity to model parameters**: Agent-based models are often sensitive to parameter tuning, necessitating careful experimentation to identify suitable values.* ***Training time and Optimization**: Training intelligent agents using machine learning is time-consuming, with the duration varying based on the number of agents and the complexity of the model.*
3d. Have you integrated other modelling techniques (e.g., machine learning, network analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.	*Yes, we have integrated machine learning techniques, specifically Unity ML Agents, with our agent-based modelling and simulation. Unity ML Agents is a powerful framework that enables us to train and deploy intelligent agents using the Proximal Policy Optimization algorithm. By utilizing reinforcement learning, we can train agents to learn optimal behaviours, allowing them to make informed decisions based on rewards and penalties. This adaptive learning process enables the agents to continuously improve their actions over time, resulting in more realistic and dynamic simulations.*
3e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.	*We are employing methodologies that focus on optimizing policy functions using the Proximal Policy Optimization algorithm. These methodologies are:* ***Reward Shaping**: This methodology designs and shapes the reward function to guide the agent's behaviour while training. It provides rewards and penalties based on the desired behaviour and intermediate goals.* ***Observation Sharing**: Agents in this methodology can share partial or complete observations with each other. This enables them to learn from the experiences of other agents and improve their individual and collective performance.* ***Exploration and Exploitation**: Agents in this methodology need to explore different strategies and learn from their interactions with other agents and the environment while simultaneously exploiting their learned policies to maximize their individual and collective reward.* ***Multiple Epochs and Mini-Batches**: This methodology involves splitting the collected data into mini-batches that are used for updating the policy over multiple epochs. This*

	approach enhances stability and efficiency in agent learning. ***Joint Reward Design****: This methodology focuses on designing the reward function to enable agents to share learned policies. It encourages collaboration and discourages unnecessary competition between agents.* ***Centralized Critic, Decentralized Actor****: This methodology utilizes a shared critic network to estimate the value function based on the collective action of all agents, while each agent maintains its own actor network for determining individual actions.* ***Documentation****: We document our project, including model specifications, assumptions, parameter values, and experiment configurations.* ***Verification and Testing****: We verify and test our models and simulations to identify and correct any implementation errors or unintended behaviours. This process includes unit testing and scenario testing to ensure the model behaves as intended across various scenarios.*

3.4 LIU

1. General Information:

Question	Answer
Organization/Company	*Linköping University (LiU)*
Edited by	*Ludvig Knöös Franzén, Jorge Lovaco*

2. System-of-Systems Engineering Methodologies:

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*We at Linköping University (LiU) have an envisioned holistic SoSE methodology that we follow in research related to the topic. This methodology was first introduced in* [1] *and further expanded upon in both PhD theses (* [2], [3]*) and licentiate theses (* [4], [5]*). Essentially, the methodology outlines how an SoS can be analysed at different levels, ranging from SoS needs, capabilities, SoS architecture and design, constituent system design and sub-system design.*

	More specifically, LiU has used the following methodologies in SoS related research: • *Ontology with reasoning capabilities: To represent and perform design space reductions for SoSs.* • *Agent-based Simulations: To evaluate different SoS solutions and their performance in various scenarios.* • *Visual Analytics: To present SoS analysis results in interactive dashboards in order to provide facilitated decision making.* • *Architecture Frameworks: In order to understand the many viewpoints of an SoS and to understand how concepts such as "Needs", "Capabilities" and "Functions" are related to each other.* • *More advanced regression analyses: Such as Singular Value Decomposition and Symbolic Regression with Genetic Algorithms to estimate new system solutions on the CS and sub-system levels.* • *Multi-disciplinary Design Optimization (MDO) to evaluate and optimize SoS and CS solutions.*
2b. What tools do you use in SoSE?	*We use/have used a variety of tools connected to SoSE. The main tools utilized are listed below:* • *NetLogo (Agent-Based Simulations tool).* • *Protégé (OWL Ontology editor that supports reasoning).* • *Power BI (A tool for creating interactive dashboards and result charts).* • *UML/SysML (in Architecture Frameworks, such as the Unified Architecture Framework (UAF)).* *We have also utilized many other tools and languages in our analyses, for example: MATLAB, Excel, Python, ModeFrontier and more.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	*Yes, our methodology from* [1] *is still being investigated and realized in different ways. There are believed key enablers that can support the various levels of the method, but this must be further investigated in the future. For example, designing constituent systems is quite well known with well-established methods. However, the operational aspect of SoSs and how their concept of operations (ConOps) should be analysed and designed is still uncertain. The same is also true regarding the impact of ConOps on SoS architectures.* *Also, for our simulations: Computational power is always something that can be better…*

2d. What knowledge do you have of current state-of-the-art (SOTA), advantages, disadvantages of different methodologies and their applicability to your organization/company?	*The advantage of the methodology from* [1] *is that it considers a holistic view of an SoS and focuses on understanding the problem and available design space of solutions. Consequently, it does not focus on a single best solution against a fixed set of requirements, but rather on the solutions that can deliver the needed capabilities over time.* *Architecture Frameworks are very good for understanding the many viewpoints of an existing SoS, however, they are less useful when it comes to the design of a new SoS. The structure of the frameworks themselves can however be exploited for this purpose as shown in for example* [6]. *Ontologies are very flexible to use as a modelling tool for SoS entities and their relationships. However, utilizing ontologies comes with a steep learning curve. Additionally, OWL ontologies are limited in terms of numerical calculation capabilities.* *Finally, the operational parts of SoSs (for example: how to approach ConOps at an SoS level) have no clear "design" methodologies as far as we know. This is a gap that the COLOSSUS project hopefully can cover to some extent.*
2e. Are there some best practices in SoSE that you know of? If yes, which?	*Yes, there are several books and documents that describe "best practices" for SoSE. Some examples are "System of Systems Modeling and Analysis"* [7] *and "Systems Engineering Guide for Systems of Systems"* [8]. *Traditional SE approaches and practices are also not to be discarded when doing SoSE.* *It is always a good idea to have the "Maier characteristics"* [9] *in mind when doing SoS related Analyses.*
2f. Which of your (if any) listed methodologies do you think are of most relevance within the COLOSSUS project and why?	*We believe that our methodology from* [1] *together with the guidelines in* [7] *and, for example, the "Mission Engineering Guide"* [10] *could provide a very relevant view and understanding of the upper levels of holistic SoS analyses, like the ones to be performed within the COLOSSUS project.*
2g. Are there any SoSE methodologies, or similar, that you are aware of but would like to investigate in more detail under the COLOSSUS project?	*Yes, as mentioned before, the suggested methodology in "System of Systems Modeling and Analysis"* [7] *would be interesting to apply when architecting new SoS solutions to be evaluated.*

3. Agent-Based Modelling and Simulation Experience:

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*We have utilized ABM/ABS in SoS studies for both search & rescue as well as wildfire fighting operations and scenarios. We have used them for System of Systems Engineering and generation of requirements for Aircraft Conceptual Design. We have studied different ways of creating RCE workflows with the ABS for extracting desired Capabilities depending on different Needs. We have studied the influence of Fidelity levels in ABM&ABS. Also, we connected ABS with Python code to improve the analysis.*
3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	***NetLogo and Agents.jl*** *NetLogo is a well-established and known tool for building Agent-Based simulations. The software has existed for a number of years being very stable and with extensive documentation and examples available. The language is very easy to use and building interfaces is rather easy with its drag-and-drop options for doing so. It is built using Java, which ensures that it will run on almost any computer. It allows the user to run parallel simulations and perform behavioural searches. It is Python compatible, which allows for extra options. On the other hand, being a Java based tool, the computational performance is not high-end, being noticeable for models running more than 200 000 agents at the same time.* *Agents.jl is a package for the language Julia. Julia is known to be a very fast language, with strong interoperability with other languages such as Python, R, C and C++. The package allows to build simulations with animations and use maps from OpenMaps. On the other hand, the Julia language is still very young, so the language itself changes rather often, leading to a high maintenance when developing code. The package itself seems not to see an extensive use and maintenance, which leads to a lot of ABM code maintenance needed for just some more performance. The package is also pure code based, in other words, building a user interface for it will need full development efforts.*
3c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?	*Computational time and software compatibility. Even though the results can usually be stored in a form that can be convenient, it will always need some kind of "curing". The tools themselves are hardly ever compatible with other tools/languages.*
3d. Have you integrated other modelling techniques (e.g., machine learning, network	*Yes:*

analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.	1. *We have coupled our ABM/ABS to ontologies that essentially describe the agents to be involved in different SoSs to be investigated.* 2. *We have coupled ABM/ABS with RCE workflows.* 3. *We have coupled ABM/ABS with Python code for designing flight trajectories, clustering data and running FMUs (digital twins) together with the simulations.*
3e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.	*No, not any specific practices. We mostly try to approach it with an SoS perspective though.*

4. **Miscellaneous:**

Question	Answer
4a. Do you have any suggestions/recommendations on relevant literature for State-Of-The-Art in connection to SoSE?	*Yes, for example:* • *"System of Systems Modeling and Analysis"* [7]. • *The "Mission Engineering Guide"* [10]. • *Architecting Principles for System-of-Systems* [9].

3.5 ONERA

1. **General Information:**

Question	Answer
Organization/Company	*ONERA*
Edited by	*Frederic Georges, Olivier Poitou*

2. **System-of-Systems Engineering Methodologies:**

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*Even if we do not have an integrated view of the full process of SoSE, some parts have been explored:* *ONERA worked on the key performance indicators that could reflect the performance of a socio-technical system of systems, a so-called coalition in the chosen application. This application was a CSAR (Combat Search and Rescue) scenario. Related publications are:* [11], [12].

	ONERA also has played the role of building a full model of an SoS, namely the aeronautical maintenance system, in several projects in collaboration with Airbus (no exploitable references unfortunately) and, later, in the AIRMES European project. A related publication is: [13]. *More recently ONERA also explored the process to build a unique formal representation of a complex system, or a system of systems, from many imperfect contributions from stakeholders. This, mainly theoretical, work focuses on identifying the different roles existing in this process, in particular the main one, referred to as the transcriptor in our work. Then identifying the actions associated with each role. Properties to evaluate the quality of an in-progress formal representation are also defined as they cannot be exactly the ones expected from a "simple" system model. A publication associated with this work is unfortunately only available in French for the moment:* [14]. *To support the internal experimentations about the previous subjects, we often rely on an internal prototype of modelling/metamodeling tool, introduced in the following articles (also mentioned in the "Formal architecture modelling..." article forementioned):* [15], [16]. *In those works, we often refer to the free modelling and model federation approach from this team (external references):* [17], [18]. *ONERA also explored coupling MBSE with MDAO on the illustrative example of a drone having as mission to check a high voltage line, related publications are:* [19], [20]. *A common point that a drone system has with a system of system, is the central notion of mission. The mission has to receive a new particular (modelling) care because its correlation with the "product under study" is weaker that in a traditional system development process. In the case of a drone system, or a system of systems, the same component systems, or even the same system, may have to deal with several missions over time. On the opposite, a given mission may be achieved by different systems. So, the system and the mission do not have the same life cycle. Though, an "association" between a given system and a given mission must be evaluable, to check at least for adequacy (this system is able to fulfil the mission) but also for different dimensions of performance (cost, time, resources...). ONERA started to investigate this point, some very preliminary information may be found in:* [21].

	External references about this work also include GRL a goal representation language we used to identify stakeholders, their expectations and how well they may be fulfilled ([22]*) and the dedicated operational mission architecture framework (OMAF):* [23], [24].
2b. What tools do you use in SoSE?	***SimLab**: We develop custom simulation tools in C++ in order to perform system of systems (SoS) studies (for instance, SoS performance evaluation). However, SimLab is not integrated into an actual SoSE workflow or methodology. SimLab architecture allows for multi-actor simulations and could be used for agent-based simulations, but proper AI/autonomous systems are not implemented. Systems behaviours are modelized using behaviour trees.* ***Knowledge and Reasoning representation with an internal tool prototype (Weird)**: Mainly focused on building a consensual formal system architecture description from many partial, heterogeneous and imperfect contributions. (static) Properties are expressed in a "kind of" multi-sorted first order logic (can be seen as halfway between Alloy and a classical SysML description, and not so far from an Ontology by some aspects). Similar to SimLab, this work is not integrated in a smooth SoSE workflow.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	*Stakeholder contributions to system architecture description (stakeholder knowledge) are often provided in graphical representations, produced by tools that are not oriented towards building a formal specification. The work to integrate them in a unique formal description (knowledge base with good properties) is still very manual and takes a lot of experimentation time.*
2d. What knowledge do you have of current state-of-the-art (SOTA), advantages, disadvantages of different methodologies and their applicability to your organization/company?	*Not really a methodology, though some guidelines are found in the documentation, but Architecture Frameworks, like the NATO Architecture Framework, at least help to figure out what may be described to document an SoS, and suggest an organization for the documentation of the different artefacts, views and viewpoints.*

3. Agent-Based Modelling and Simulation Experience:

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*We have experimented a little on ABM/ABS in another project to evaluate different modelling of opinion propagation among populations.*

3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	*We used NetLogo, but not intensively enough to have a clear view on this question. Mixing symbolic rules and numerical indicators was rather a headache, but this may be due to our poor knowledge of the tool. Same feeling about quickly encountering some scaling issues, which may be due to incorrect usage.*

3.6 TUD

1. General Information:

Question	Answer
Organization/Company	*Delft University of Technology (TUDelft)*
Edited by	*Marta Ribeiro*

2. System-of-Systems Engineering Methodologies:

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*TUDelft does not have practical experience with employing a system of systems approach.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	*TUDelft can benefit from learning about the advantages of employing a system of systems modelling.*

3. Agent-Based Modelling and Simulation Experience:

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*TUDelft uses agent-based modelling (ABM) as a base for its research. ABM is used to simulate the end-user system (e.g., airport environment, airline operations, advanced air mobility), and test the impact of new operations and optimization algorithms.* *ABM techniques have been employed by TUDelft in several European Projects. The following recent projects are highlighted:* • *The EU Horizon2020 project ORCHESTRA on coordinating and synchronizing multimodal transport – ABM is used to simulate the motion of each transport vehicle.* • *The EU SESAR AEON project on electric green airport surface movement operations – ABM is*

	used to simulate and test multi-agent path planning. *Recent examples of published research focusing on agent-based modelling:* [25], [26]. *These assess the effects of multi-agent influences on airline performance.* *Additionally, the course "Agent-based Modelling and Simulation in Air Transport" is offered by TUDelft at the beginning of the Sustainable Air Transport MSc program, teaching students how to formulate a practical air transportation problem as an agent or a multiagent system model.*
3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	*TUDelft has developed multiple agent-based, open-source simulations. A mature example is the Bluesky – Open Air Traffic Control Simulator tool. This is a fully open-source and open-data approach to air traffic simulation. The biggest strength of this tool is its large application range – it can be used both for manned and unmanned aviation and can represent a large variety of aircraft and operational environments. This tool has been recently used as the simulation environment in the following European Projects:* • *The EU SESAR Metropolis 2 project on enabling air traffic in high-density urban environments.* • *The EU SESAR Bubbles project targeting the formulation and validation of a concept of separation management for UAS in the U-space.* *Within a teaching context, TUDelft has also used NetLogo, a multi-agent programmable modelling environment with its own API. The simplicity of this tool is a great advantage for teaching, nevertheless it also limits its applicability.*
3c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?	*Correct representation and simulation of all the interactions in a multi-agent system is a time-consuming task. It is important to pre-define the range of actions/operations in advance. Otherwise, it may be that, at the end of project, the simulation platform is not robust enough to be used as a validation tool.*
3d. Have you integrated other modelling techniques (e.g., machine learning, network analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.	*Yes, employing machine learning to improve air traffic operations is one of the main interests at TUDelft. We have published work using machine learning directed at air transportation:* • [27]: *Reinforcement learning (RL) was used to guarantee a minimum safety distance between UAVs in an urban environment. The RL model can*

	control the actions of each UAV to prevent future conflicts. • [28]*: Supervised learning was used to predict future flight delays.* *TUDelft has built state-of-the-art supervised and reinforcement learning models. Finally, a current research line is how to make the results of such "black box" models more explainable and interpretable to increase acceptance and understandability by the end-user.*
3e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.	*TUDelft employs the following standard practices:* • *The Overview, Design concepts, and Details (ODD) protocol for describing Individual and Agent-Based Models: A framework for defining a model in a way that ensures that any other modeller can rebuild the model and replicate the results. This is a good practice which informs the modeller how to properly describe an agent-based model. More information can be found in* [29]. • *Object-Oriented Programming: Almost all ABMs are built using an OOP language. An object is able to store data and has methods that determine how it processes these data and interacts with other objects. In this way, each agent can then be represented as an object.* *Moreover, simulation tools by TUDelft are written in Python. It is popular, free, and open source. Additionally, Python programs can easily be used on multiple platforms (Windows, Mac, Linux).* • *Stochasticity: Any modeler must take into account that a large number of simulation runs are necessary to generate a representative result. The magnitude of uncertainty can only be understood by running multiple repetitions of a particular scenario.* • *Cross validation: For validation, the model output must be compared to previously validated models and, if possible, to real data.*

3.7 UNINA

1. General Information:

Question	Answer
Organization/Company	*University of Naples Federico II (UNINA)*
Edited by	*Michele Tuccillo, Pierluigi Della Vecchia, Fabrizio Nicolosi*

2. System-of-Systems Engineering Methodologies:

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*Application of model-based design techniques to the system level (the aircraft) and its components, more than the system of systems level, during both conceptual and preliminary design stages.*
2b. What tools do you use in SoSE?	*Matrix Diagrams and Quality Functional Deployment Matrix to determine the weight of a specific design discipline (at the aircraft level) with respect to given requirements.*
2f. Which of your (if any) listed methodologies do you think are of most relevance within the COLOSSUS project and why?	*Our methodologies (not listed) are addressing at System level and they are relevant in the design and optimization at system level.*

3.8 VTI

1. General Information:

Question	Answer
Organization/Company	*VTI*
Edited by	*Chengxi Liu*

2. System-of-Systems Engineering Methodologies:

Question	Answer
2a. What methodologies do you use in System of Systems Engineering (SoSE)?	*For me, the system I deal with is the transport system. The transport system consists of:* *1. Individuals and/or companies who have a travel demand, i.e. they want to travel.* *2. Vehicles that carry individuals as a means of transport.* *3. Transportation network for each mode where the given vehicle is allowed to be used.*

The transport system can then be considered as a subsystem if one would like to analyse the broader impact of relevant policy measures. For instance, the transport system can be coupled with the land use system (where the city would like to build new facilities that further attract people to travel there). Transport systems can also be coupled with economic systems where the productivity of a certain type of goods can be associated with the accessibility at which the goods are produced.
The transport system can also be viewed as a system of systems where the subsystem consists of individuals, vehicles and networks. Like in COLOSSUS, each vehicle is then a complex system. However, my expertise is mainly in the analysis of transport systems where the subsystem, e.g., vehicle is simplified as an "object" with several relevant attributes, such as vehicle length, vehicle maximum speed, vehicle capacity, etc. In that sense, the micro system of the vehicle is exogenous and simplified, presented by the given attributes.

The traditional methodology to analyse the transport system is the so called "four-step model" which has been widely used in transportation engineering since the 1970s [30]. *The methodology includes the estimation of travel demand and demand-supply interaction, answering the following questions:*

1. *How many trips do we generate from a given area (trip generation)?*
2. *How many trips travel from A to B (trip distribution)?*
3. *How many trips travel from A to B by a given mode (mode choice)?*
4. *What is the travel time from A to B by a given mode, given the fact that if many choose the same route then there will be congestion and thus longer travel time (route assignment)?*

There have been several advances in furthering the "four-step model" method. One direction is the further understanding of travel demand, i.e. trip generation, trip distribution and mode choice (questions 1-3 in the list above). The advance includes better models/tools for each of these questions. For instance, there are new novel models that take into account people's attitudes, social norms, considering bounded rationality in making choices.

	And these models provide better accuracy and interpretability in answers to these questions. Another direction is the further understanding of demand-supply interaction (interaction between the questions 1-3 and the question 4 in the list above) where the main advances result in so called dynamic traffic assignment [31]. *From a system of systems point of view, like in COLOSSUS, the vehicle in the traditional method is considered in the following way:* 1. *The vehicle specific attributes may play a role in the mode choice and destination choice. In many four-step models nowadays, there is often a sub-model called car ownership model that models which car a given individual will buy or rent, i.e. to model the whole car fleet.* 2. *The vehicle specific attributes may act as a constraint in route assignment. For instance, if the bus has a low capacity while many travellers choose to take the bus then the bus will be very crowded and thus some bus travellers may travel by other means of transport instead in the next iteration.*
2b. What tools do you use in SoSE?	*The four-step model is being realized in various software which are widely used by many municipality planning units all over the world. The ones I use are PTV VISUM* [32] *and TransCad* [33]*. I also made my own tailored four-step model in MATLAB and Python for use in some research projects.*
2c. Are there any "gaps", or other aspects that you feel are missing, in the methodologies and tools that you are using currently (for example, things that would make your work more efficient or effective)?	1. *There is in general a lack of standards of different data formats that each software can import. A lot of time is required to convert agent, network and vehicle information into a readable format for the given software.* 2. *It is not easy to configure the software. Thus subsystems, such as individual vehicles, can only be presented by a set of predefined attributes. Although custom user-defined attributes are in general allowed which provides the possibility to further connect the tool with other external models/tools. However, some design principles may be taken so that the model can produce results that can be exchanged via API-like interfaces.*

2d. What knowledge do you have of current state-of-the-art (SOTA), advantages, disadvantages of different methodologies and their applicability to your organization/company?	*VTI has a broad knowledge and tools, including driving simulators, test facilities for road materials and vehicles, Virtual Reality and many more. Each may focus on a specific system. Unfortunately, my knowledge of these tools is so far very limited so I cannot really assess SOTA, advantages and disadvantages, except that one disadvantage is that there are rarely joint research initiatives to use different tools together (however, I understand that this will become more available in the future).*
2e. Are there some best practices in SoSE that you know of? If yes, which?	*There has been a long effort in transportation engineering to couple the transport system with the land use development (one example in the UK:* [34]*). There are many research papers and models that developed co-evolutionary methodologies where the accessibility of travel affects where people live and work and vice versa. In that sense, the transport system and land use system are both subsystems of the city as a whole system.*
2f. Which of your (if any) listed methodologies do you think are of most relevance within the COLOSSUS project and why?	*I think that the "four-step model" methodology to analyse transport systems is very relevant for the ADAM use case in COLOSSUS and work package 5. The "four-step model" can be used to model regional travel but also long-distance travel, but the use of this methodology to analyse aviation mobility is still underexplored. Its strength is its ability to analyse different transport modes jointly where the attractiveness and constraints of each mode can be properly represented in the "four-step model" framework.*
2g. Are there any SoSE methodologies, or similar, that you are aware of but would like to investigate in more detail under the COLOSSUS project?	*The "four-step model" methodology is one candidate, but of course, there are many other options.*

3. Agent-Based Modelling and Simulation Experience:

Question	Answer
3a. What experience do you have in agent-based modelling (ABM) and agent-based simulations (ABS)?	*Mobility simulation is one type of agent-based simulation (ABS) platform that is specialized for analysing individual mobility. Mobility simulation is closely related to the four-step modelling methodology described above but differs in various ways. It also answers the above four questions but is not limited to those. Four-step modelling does not model individual travel (i.e. how many trips does one travel) but models trips generated from a given zone, while mobility simulation explicitly models individual travel. Furthermore, mobility simulation models not only show how a given individual would make each of the travel*

	choices as described above (mode, destination and number of trips) but also when a given individual would travel and how he/she would organize the travel, for instance, first go to supermarket then go to office or the other way around. *The main difference between four-step modelling and mobility simulation is that mobility simulation explicitly tracks the individual movement in a network on each time step while four-step modelling is a "static" model, meaning it does not care about where the trip is at a given time step, instead, it only cares about whether the trip has been at this particular location (link or node) but the researcher does not know when.*
3b. Which ABM/ABS software have you utilized and what are their strengths and weaknesses?	*I use existing mobility simulation: MATSim* [35] *(open source) and Transmodeller* [36] *(commercial), although we currently do not have the license. Transmodeller is only used in some practical, consultant projects. For research, I use mainly MATSim.* *The strength of MATSim is that it is designed to be highly configurable, allowing you to replace whatever component in the simulation by which a user-defined variant. The simulation platform is extendable and provides a high degree of freedom for the user to invent their own component and then "plug-in". This is of course with a sacrifice of not being very user-friendly in the sense that the user must have a deep Java programming language knowledge or at least a basic understanding of Google Guice* [37]. *This also makes debugging hard since the error message can only be understood by persons with good Java knowledge.*
3c. Are there any specific challenges or limitations that you have encountered while using ABM/ABS in your work or research?	*So far, the limitation with MATSim is that it is heavily focusing on tackling congestion problems or any capacity problem in the network, that takes most of the computational time. When the use case is less related to capacity problems, MATSim's capability feels more like a "mismatch" of being the right tool to solve the problem.*
3d. Have you integrated other modelling techniques (e.g., machine learning, network analysis, ontology) with ABM or ABS? If yes, please provide examples and elaborations.	*Network analysis is by default "built-in component" MATSim ABM. Machine learning is more exogenous. MATSim can produce output that is being read in a machine learning tool and then get the feedback.* *One example, or use case, can be Machine Learning (ML) based traffic signal control. MATSim reads in a traffic signal plan that is generated by ML and then MATSim*

	produces the average waiting time at each intersection that feeds into ML to generate a new signal plan.
3e. Do you use any standard practices, methodologies or guidelines when developing ABM/ABS? If yes, please specify.	*We have extensive experience although I could not say we have a "standard" practice. But our procedure of developing scenarios in MATSim is very much standard in some sense. In general, it includes:* 1. *Understanding of the underlying problem and identifying which modification (component) is needed in the default MATSim setting.* 2. *Implementation of user-defined components, sometimes, requires writing a whole new add-on to "plug-in".* 3. *Test of components in a test simulation scenario (toy network).* 4. *Preparation of all necessary input files, like agents' travel plans, network, and vehicle files.* 5. *Run the simulation.* 6. *Post-processing of results.*

3.9 Summary of Collected Questionnaire Answers and Identified Gaps

The collective answers from the distributed questionnaire give an overview of the current knowledge in holistic SoSE and ABM/ABS within the COLOSSUS project. However, this section has also highlighted some of the existing gaps in partners' methods and best practices. The most prominent of these identified gaps are listed below:

- Most distinguishing is the current lack of actual implementations of truly holistic SoSE methodologies. There is quite extensive knowledge in various related parts, which often are domain specific, but a general methodology that formally unifies all different aspects is missing.

- Methods and processes for creating and exploring full-on SoS architectures are non-existent. There are plenty of approaches for describing SoS architectures, but no actual SoS architecting processes. Inspiration from, for example, architecture frameworks should however be taken in a yet-to-be developed architecting approach for SoSs where different designs can be explored.

- Approaches that cover the design for behavioural aspects (or the collaborative part) of SoSs are missing. ABM/ABS is believed to be the key for investigating behavioural aspects and emergent behaviours of SoSs, however, there are no guidelines for how such aspects should be designed and explored.

- Many partners contribute extensive knowledge in ABM/ABS for SoSs. Nevertheless, formal methods and practices for SoS representations are lacking, especially from the collaborative and behavioural aspects.

Best practices for specifying SoS simulations that enable large scale emergent behaviours from interactions among involved systems are yet to be defined. However, machine learning techniques are believed to play a large role in such ventures.

- Finally, verification, and validation for ABM/ABS results become more challenging with the scale and complexity of SoS analyses (which also is true from a computational demand point of view). How such large-scale analyses can be verified and validated yet remains to be determined. However, well-established methods, such as those presented in [38], can still be used to some extent.

Despite identified gaps, the involved COLOSSUS partners bring together a wealth of knowledge across various domain-specific areas. This collective expertise significantly contributes to the holistic SoS analyses envisioned for the project. This is particularly evident when examining the constituent- and subsystem perspectives, where a plethora of methods, tools, and best practices are available, with a special emphasis on aircraft design. Many partners also have extensive prior backgrounds in SoSE and related fields. Combined, they form a robust foundation for addressing the identified gaps within the scope of the COLOSSUS project.

4. LITERATURE STUDY

This section presents the results from the two-part literature study performed.

4.1 Holistic System-of-Systems Analyses and Engineering

In the realm of System of Systems Engineering (SoSE), a comprehensive and detailed literature study reveals a multifaceted approach to understanding and addressing the complexities related to designing, implementing, and managing interconnected systems. This review synthesizes insights from key scholarly works, each contributing unique perspectives and methodologies to the holistic engineering of System-of-Systems (SoS).

4.1.1 Fundamental Concepts and Definitions

At the forefront of this discourse is the work of Mark W. Maier's [9]. His work discusses the concept of SoS, which comprises large-scale components that function both as independent entities and as part of a larger whole. The paper aims to establish a clear taxonomic distinction for SoS, differentiating them from other complex systems based on the operational and managerial independence of their components.

Maier proposes a two-part core definition of SoSs. Firstly, the components of an SoS must be capable of fulfilling valid purposes individually. Secondly, these components should be managed, at least in part, for their own objectives, not just for the purposes of the entire system. This definition challenges the traditional criteria, such as complexity and geographic distribution, that are often used to classify SoSs.

The paper also discusses design principles, heuristics, best practices, and patterns specific to SoSs. These principles are crucial for understanding the distinct approaches needed for effectively architecting SoSs. Maier's work suggests that SoSs require unique considerations in their design and management, significantly different from those applied to more traditional systems.

Building upon Maier's foundational work in establishing a distinct definition of SoSs, which emphasizes the operational and managerial independence of their components, Meentemeyer's [39] comprehensive exploration of the SoS concept discusses deeper into the myriad of definitions that exist in the realm of SoSs.

Meentemeyer's presents a detailed exploration of the concept of SoS and the discipline of SoSE. The paper acknowledges the lack of a universally accepted definition for SoS, typically referring to large-scale or complex systems providing a set of capabilities. The paper navigates through over 40 different SoS definitions, with common themes including operational and managerial independence, geographic distribution, emergent behaviour, and evolutionary development. Focusing on the Boardman-Sauser model [40], the paper outlines five distinguishing SoS characteristics: autonomy, belonging, connectivity, diversity, and emergence. These characteristics provide a framework for understanding the nature and behaviour of SoSs, offering insights into their complex dynamics. Discussing SoSE, the paper emphasizes the differences between traditional SE and SoSE. It describes SoSE as a discipline that requires a broader scope, dealing with more complex efforts, and involves high levels of uncertainty due to the dynamic nature of SoS environments. Key differences include the need for dynamic reconfiguration of architecture, interoperability of component systems, and a focus on emergent behaviours. From the perspective of various stakeholders, including users, developers, trainers, testers, sustainers, acquirers, and researchers, the paper contrasts the approaches of traditional SE and SoSE. It highlights the evolving requirements, distributed development, and the need for constant adaptation in response to changing needs in SoSE. The paper suggests nine principles and practices for guiding individuals working on SoS projects. These include the use of integrated product teams, global risk management, decision analysis, emphasis on interoperability, interface management, utility of modelling and simulation, understanding of emergence, agile development concepts, the use of "systemigrams" for SoS description, and paradoxical thinking.

Lastly, the INCOSE System Engineer Handbook [41] introduces a nuanced layer of challenges regarding the development of SOS. It highlights the independent operation of system elements, each being operational on its own, thus adding layers of complexity in their integration and management. The handbook is aligned with the ISO/IEC 15288 standard [42] which describes SE and system life cycle processes. The handbook emphasizes the diversity in life cycle of SoS components, spanning from the development to operational stages thus presenting challenges in synchronizing these disparate life cycle stages. Ambiguity in initial requirements is another critical challenge cited, requirements for SoS tend to evolve and gain clarity as the individual system elements mature. The complexity increases non-linearly with the addition of system elements and such is identified as a major challenge. Managing and integrating various system components into an SoS increases disproportionately as more components are added. One element that contributes to the complexity is conflicting interface standards where data exchange and interactions between different system elements becomes a major hurdle.

The Handbook also highlights the concern of autonomous management overshadowing the engineering aspects in SoS due to the complex boundaries where the lack of clear definition and control over the external interfaces of the system elements can lead to operational inefficiencies. SoS is a perpetual process, necessitating ongoing adaptation to technological advancements and changes in the life cycles of the constituent systems.

To address the challenges, the handbook suggests a blend of SE approaches that combine both the systematic and procedural aspects described alongside holistic, nonlinear, and iterative methods.

4.1.2 System-of-Systems Engineering Methodologies and Approaches

Expanding from the foundational definitions and characteristics of SoS as explored by Maier and Meentemeyer and focusing on the practical applications and methodologies in SoSE, Mo & Beckett [43] discusses the intricacies of engineering and operating SoSs. The work emphasizes the necessity for advanced modelling and analysis techniques, which are essential in developing a comprehensive system that meets essential requirements. This focus is particularly relevant in the context of complex socio-technical systems where engineering decisions have far-reaching implications on societal and technological aspects. The work serves as a cornerstone for professionals engaged in the intricate task of integrating diverse engineering systems, offering a pragmatic lens through which the operational challenges of SoS can be understood and addressed. The authors propose using the standard ISO 42010, specifically the chapter titled “Systems and software engineering—Recommended practice for architectural description of software-intensive systems” [44] where a framework to characterize large, complex systems as well as the components of such systems is defined. Though the standard is not aimed at an SoS level, it provides insights to good practices for defining a system, it also serves as a guideline of a framework or set of practices for larger and complex systems. As a first step, the author proposes the use of an 'Element Interaction Matrix', a tool designed to map out the relationships between different components of a system. It has proven to be insightful, revealing a greater number of interactions and connections than initially anticipated. This approach highlights the complex interdependencies within SoS. As a second step, defining stakeholders, architecture description and Rationale. Stakeholders may be classified according to the role or assignment they are responsible for, either by involvement in a different engineering or operational phase. Defining the architecture description should consider if the experiences in a knowledge structure are captured, provides a baseline to build new structure or content and represents the minimum expectations. In the rationale domain, it is highlighted that the standard requires evidence of the consideration of alternative architectural concepts, and the rationale for considering these alternatives. The authors suggest a critical questioning approach and a knowledge-oriented approach.

The last step involves a thorough consideration of the mission and its functionalities, alongside the system's architecture with the goal of ensuring that the system architecture is fully representative and aligns with the expectations and requirements of each stakeholder.

Jamshidi [45] discusses the principles and applications of SoS engineering. Jamshidi’s contribution lies in conceptualizing SoS as a conglomerate of individual, possibly heterogeneous systems, integrated to enhance overall robustness and reliability. This theoretical foundation is instrumental in framing SoS engineering not merely as a technical endeavour but as a holistic practice that intertwines various functional systems. It provides a broader understanding of how individual system functionalities, when harmoniously integrated, can lead to a more robust and cost-effective overall system.

While considering the practical challenges, the work from Staack et al. [1] discusses the complexities of SoSE in aerospace, emphasizing a global approach to product development. This approach accounts for both technical and non-technical factors, including economic, managerial, and regulatory aspects, acknowledging the interconnectedness of products with their operational environments. In defining SoS, the work highlights characteristics like operational and managerial independence of components, geographic distribution, emergent behaviour, and evolutionary development. These features distinguish SoS from traditional complex systems. However, SoSE as a scientific discipline lacks a fully defined structure, comprehensive guidelines, or best practices for the entire design process, underscoring the necessity for an interdisciplinary approach. The paradigm shift in SoS engineering moves towards virtual simulations and model integration, employing frameworks like multidisciplinary design optimization (MDO). This shift necessitates the inclusion of non-mechanical engineering domains and an interdisciplinary systems engineering approach. The document proposes a phase-based process for holistic product development within the SoS context, aiming to align needs, capabilities, and system requirements in the initial stages. This process encompasses five main levels of interest, ranging from needs and boundary conditions to subsystems design space. The proposed SoS approach extends beyond traditional design methodologies, placing more emphasis on a thorough system capabilities analysis and integrating various factors, including geopolitical and environmental aspects. This is a departure from conventional design-to-cost or design-to-value driven methods. Various methods and techniques are identified as key enablers for realizing this holistic approach. These include meta-modelling, common language, matrix-based approaches, relational/graph-based methods, forecasting methods, value-driven and robust design, epoch analysis, data-driven design and trade space exploration, visual analytics, and big data. Each of these plays a crucial role in different phases of the holistic design engineering process.
However, the complexity and diversity of modelling approaches in SoSE present significant challenges. The lack of an established holistic SoS research field and the varying backgrounds of research groups add to this complexity. The paper discusses the necessity of integrating different domain experts into a cohesive framework and explores the potential role of AI and big data in enhancing the design process.

The Mission Engineering Guide [10] serves as a comprehensive resource for Mission Engineering (ME), aimed at both new and experienced practitioners. It integrates technical and engineering rigor into ME analysis, focusing on problem formulation, principle understanding, and analysis in a mission context. ME is defined as a top-down approach utilizing systems and SoS in operational missions to guide capability development and address war fighter needs. The process is analytical and data-driven, beginning with a clear problem statement and mission characterization, followed by identifying key metrics for analysis. The guide emphasizes the importance of consistent mission definitions across alternative approaches and the role of metrics as quantitative measures for assessing mission or system performance. It details the examination of the interplay between operational environments, threat activities, tasks, and systems used in missions, underlining the significance of realistic assumptions and constraints in ME studies. The outputs of an ME analysis include documented results, reference architectures, and curated data models. These products play a crucial role in identifying mission capability gaps and guiding technology development, investment decisions, and future mission strategies, aligning with the demands of holistic system-of-system engineering approaches and implementation.

The work from Keating et al. [46] is expanding the discussion by referring to holistic methodologies and implementations within SoSE, emphasizing the integration of complex systems into cohesive metasystems. This field of study advocates for an all-encompassing approach that ensures these systems operate effectively within larger systems of systems. One of the critical issues in SoSE is the integration of complex, individual systems into a broader, more complex whole. This process is not just a technical endeavour, it requires a comprehensive approach to guarantee that these systems function efficiently as part of a larger, integrated system.

The literature points out a significant gap in the rigorous development of SoSE concepts. It suggests a more structured approach that merges theory, empirical research, and practical applications, fostering a robust, systemic understanding of SoSE. The authors argue for a transdisciplinary approach that extends beyond purely technical matters to explore broader SoSE challenges, especially when it comes to transforming existing systems to fit within SoSE. The literature discusses the need for new methodologies, technologies, and capabilities for effectively integrating existing systems within an SoSE framework, ensuring they perform optimally within the constraints of the larger metasystem. Evaluating and evolving SoSE presents complex challenges, especially concerning long-term sustainability and the dynamic nature of SoSE. The review underscores the necessity for focused research to develop methodologies that can effectively evaluate and evolve SoSE throughout its lifecycle. To advance both the theoretical and practical aspects of SoSE, the paper proposes the development of core discipline knowledge, including theory, laws, methodologies, and principles. The authors suggest an integrated research agenda to mature SoSE as a discipline. This includes bridging the gap between research and practice, drawing insights from related fields such as traditional systems engineering, systems theory, and addressing unique system contexts including new system design, existing system transformation, operation, maintenance, and the evaluation and evolution of SoSE. The literature argues for the design of SoSE architectures, methodologies, and technologies that transcend traditional isolation constraints. This approach is essential for achieving integrated performance within an SoSE environment, allowing for a more cohesive and effective operation of these complex systems.

Lastly, the paper from Keating and Padilla [47] addresses the need for a paradigm shift in SE to effectively handle the complexities of SoSE. It contrasts the traditional SE approach, which is adept at developing objective, verifiable requirements for technical systems, with the broader and more complex domain of SoSE. This domain encompasses not only technical aspects but also organizational, managerial, policy, human, social, and political dimensions, characterized by emergence, ambiguity, and uncertainty. Key distinctions in the SoSE problem domain include holism, complementarity, pluralism, emergence, incomplete knowledge, and fluid boundaries.
The paper offers several guidelines for SoSE requirements development. These include establishing the SoS's purpose and boundaries, setting iterative objectives, identifying stakeholders and contextual issues, defining metasystem representation, and establishing performance indices. It also advises on balancing subsystem autonomy with SoS integration, allocating resources and authority, and managing coordination, integration, and standardization within the SoS. Additionally, guidelines suggest developing requirements for dealing with emergence and maintaining the SoS identity.

4.1.3 Challenges and Evolution in System-of-Systems Engineering

A systematic mapping of SoS research literature and an analysis of SoS characterization [48] revises over 3000 papers to provide an overview of the field. It reveals a dominance of U.S.-based research with a focus on military and space systems. Despite many contributors, few researchers deeply focus on SoSE, and citation rates are lower compared to other fields. The research is marked by an increase in activity since the early 2000s, with most publications in conferences and journals. Key research areas include modelling, integration, sustainability, and interoperability. The study suggests the SoS field lacks maturity, characterized by limited systematic empirical research and an unbalanced focus on certain application areas. According to the author, 21% of the relevant paper were from the military domain, followed by 11% from the space area with also 18 different application areas mentioned, each with between 1-5%, including health car, disaster management, aircrafts, robotics, power systems, etc. Another finding is that a significant number of papers addressed the Global Eath Observation Systems (GEOSS), even though they did not relate to the SoS aspects of GEOSS, but rather to some component, algorithm, etc. to be used in that SoS. It recommends fostering community building and scientific study through high-standard international events to advance the field.

The paper from Fang et al. [49] provides a guideline for framing SoSE requirements, emphasizing the need to establish clear purposes, boundaries, objectives, performance indices, and strategies for dealing with emergent conditions. As guidelines for the Framework conceptualization, it suggests establishing the purpose for the SoS, establishing the boundaries while also considering that from an SoS perspective. The latter should be fluid and subject to changes to adapt to the evolutionary nature of SoS, defining the SoS objectives iteratively and keep them flexible to adapt to new knowledge and changing context, disruptive emergent behaviours that can jeopardize the goal of the SoS, defining the stakeholders and contextual issues as in a traditional SoS approach and finally establish a metasystem representation to understand its managerial and integration role.
As for performance requirements, the paper proposes the establishment of an index of performance and expectations from an SoS level in the sense that the previously described goals and interests of each stakeholder should be addressed accordingly.
Finally, for the design requirements, it suggests that there should be clear coordination, integration, and standardization degrees to prevent suboptimization. Identifying how the SoS will address emergent conditions considering that the SoS should adapt and evolve in response to unexpected changes, is a possible use for AI. The final guideline is to establish a Framework for SoS transformation to account for the adapted behaviours from the previously mentioned point in order to increase the life cycle of the SoS while dealing with legacy and new constituent systems.

Following Fang et al., the paper from Dridi et al. [50] examines the current state of modelling approaches in SoS engineering and discusses the various techniques adapted from conventional systems engineering to fit the needs of SoS design. Similarly to [49], the authors highlight the challenges for designing an SoS such as dynamic adaptation, and emergent behaviour anticipation adaptability. The authors advocate for the MBSE methodology as a suitable framework for designing informed architectural solutions for SoS challenges.

4.1.4 System-of-Systems Architecture and Integration

The paper from Fang [51] provides an extensive and in-depth analysis of the current state and future directions in SoS architecture selection. It emphasizes the need for holistic, adaptive, and integrated approaches in SoS engineering, acknowledging the complex and dynamic nature. Regarding the challenges and critical issues in SoS architecture selection, the author mentions the interdependency of constituent systems that can lead to cascading effects and failures, making the architecture selection process complex. The author suggests integrating the activity/functional/physical models with data and machine learning algorithms to provide a comprehensive understanding of interdependencies and to identify missing factors, thus helping the architecture selection. The autonomy of constituent systems poses a significant challenge in achieving a cohesive SoS architecture and the paper suggests methods such as individual priority guarantees, game theory and negotiation to balance the autonomy and cooperation of each constituent system. The author finally mentions how an SoS should have an evolutionary nature in which it requires architectures that can adapt and evolve. As an approach to avoid the design of an SoS that is unfavourable to evolutionary nature, the paper proposes methods like dynamic strategic planning, real options analysis, epoch-era analysis and intelligent algorithms.

Building upon Fang's work, C. Guariniello et al. [52] introduces two novel methods for assessing dependencies within SoSs: Functional Dependency Network Analysis (FDNA) and Development Dependency Network Analysis (DDNA). These techniques are designed to evaluate operability, reliability, and resilience in both operational and development networks of SoS architectures, offering insights into the complexity and interdependencies inherent in SoSs. FDNA focuses on operational networks, examining the effects of different system topologies and potential degraded functioning on network operability. DDNA, on the other hand, is applied to development networks and assesses how development time and capabilities are influenced by network topology and potential delays in the development of component systems. Both FDNA and DDNA are versatile tools that can be applied to various fields and contexts, such as aerospace systems design, industrial production, and social relationships. They are especially useful for the DoD in creating analysis and decision tools for SoS, considering the typical traits of such systems as complexity, size, and partial autonomy.

4.1.5 Innovative Models and Techniques in System-of-Systems Engineering

The paper from Dahmann et al. [53] contribution lies in its practical reimagining of SoS SE processes through the wave model, which is more aligned with the incremental and iterative nature of modern SoS development. Modern SoSs are characterized by their complexity and the interdependencies of their constituent systems. Therefore, to account for the increased complexity, the author suggests a departure from the traditional trapeze model to the wave model. The wave model consists of multiple overlapping iterations of evolution, continuous analysis, and input from the external environment. The paper suggests the transition between models for representing SoS SE core elements and their interrelationships since according to the authors, the wave model provides a more useful guideline to practitioners looking for an implementation approach on their SoS program. It emphasizes the challenge in traditional SE approaches due to the independence of systems comprising an SoS. This independence complicates the ability to define boundaries and requirements clearly. The wave model proposed in the paper breaks down the SoS SE process into six time-sequenced steps, offering a more familiar and manageable approach for practitioners. This model reflects the attributes of SoSs, acknowledging the need for multiple overlapping iterations of evolution and ongoing analysis to address the dynamic nature of SoSs and its context. It is a departure from traditional systems.

While still on the wave model, the paper from Acheson et al. [54] proposes a model that aims to explore the intricate process of SoS development. The research introduces an innovative agent-based model (ABM) to simulate SoS development. In this ABM, various abstracted entities, termed as agents, are used to simulate decisions and interactions. These agents, representing both the SoS and individual systems, operate based on simplified rules, with each having its own goals and the capability to perceive environmental changes. The systemic behaviour of the SoS emerges from these interactions and decisions.

The paper from Gideon et al. [55] discusses the increasing importance of studying systems-of-systems (SoS) in systems engineering. It acknowledges the need for a more precise definition of SoS and their attributes, emphasizing that a clear understanding begins with a detailed taxonomy of these systems. The paper proceeds to suggest a categorized approach to the SoS on acquisition (i.e., dedicated or virtual), operational (i.e., chaotic, directed and collaborative) and finally between physical and abstract.

4.1.6 Summary of Holistic System-of-Systems Analyses and Engineering Literatures

The literature study on SoSE offers a comprehensive analysis of various approaches and methodologies in the field. It synthesizes insights from several works, each contributing unique perspectives to the holistic engineering of SoS. The review covers foundational concepts and definitions, methodologies, and approaches from different authors, discussing the intricacies of engineering and operating SoSs, principles and applications of SoSE, complexities in SoSE, and the need for a holistic, transdisciplinary approach. However, even though each author proposes a different approach, there is a consensus that the field has a long way to research. SoS encompasses a wide range of domains, each with unique requirements and challenges, making a one-size-fits-all approach impractical. The inherent complexity and dynamism of SoSs, where components operate both independently and interdependently, add layers of complexity to the design process. Adding more to the complexity, technological advancements continually introduce new variables and possibilities, requiring adaptive and flexible approaches. These factors contribute to the ongoing debate and diversity of methodologies in the field.

In summary, the literature study reinforces the findings from the questionnaire results in the previous chapter, reaffirming the need for further research and development in the field. It corroborates the observation that while there is extensive domain-specific knowledge, a unified, holistic SoSE methodology design remains elusive. This review also highlights the lack of formal architecting processes and guidelines for exploring SoS architectures, particularly concerning their behavioural aspects and integration of CS within the SoSE discipline.

4.2 Agent-Based Modelling and Simulation Techniques

Section 3.9 provided a summary of the gaps identified from the responses of various partners involved in COLOSSUS, covering the topics of SoSE and both ABM/ABS. From this point forward, 'ABMS' will be used to denote Agent-Based Modelling and Simulation. ABMS is believed to be a means by which to investigate SoS architectures and behaviours at an early design stage in SoSE. This section will consequently review recent and significant literature on ABMS, focusing on various related techniques and methods with an SoSE-based focus.
The methodology followed to find these relevant publications was as following:

- Browse by searching for keywords related to the COLOSSUS project, such as '*System of Systems*' and '*Agent-Based Modelling*'.
- Accessing publications through the Linköping University library and those available as open-access documents.
- Prioritizing literature with more recent publication dates, especially when they cover similar topics.

The reviewed literature will address the current gaps in ABMS and SoS. It will then be correlated with the gaps identified by the COLOSSUS project partners, aiming to highlight potential areas for future research contributions.

4.2.1 Analysis and Synthesis of Agent-Based Modelling & Simulation Applications

This subsection categorizes the relevant literature by topic.

I. Interplay of ABMS and SoS:

- Recent studies highlight the close relationship between ABMS and SoS. The parallels between SoS for systems development and Object-Oriented software approaches are explored in [56]. It suggests that ABMS provides a better understanding of SoS dynamics. A step further is taken in [57] by presenting an empirical comparison between agent-based and event-based modelling, which is important for understanding SoS analyses since the former is more suitable for exploring complex dynamics from interactions and the latter for exploring sequences and timings of events.
- Complex adaptive systems within network analyses are studied in [58]. Its alignment with ABMS methodologies illustrates the emergent behaviours in SoS. The comparison of ABMS with equation-based modelling in [59] adds another dimension to this analysis. Moreover, [60] explores theory building using ABMS versus variable-based models, providing a comprehensive view of social psychology representation through ABMS.

II. Diverse Applications of ABMS in Various Fields:

- Diverse applications of ABMS across various disciplines are collectively presented in [61], [62], [63], [64], and [65]. It is emphasized in [61] the growing recognition of ABMS in fields such as ecology and economics, showcasing its versatility. In [62], the analysis focuses on the role of ABMS in mobility transitions, with a specific emphasis on the diffusion of electric vehicles.
- Discussion is taken in [63] about the application of ABMS in modelling complex socio-political-economic dynamics, whereas [64] explores its inclusion in the study of demography and environmental sciences. Finally, [65] provides practical applications of ABMS in industrial settings, further evidencing its wide-reaching implications across different sectors.

III. Methodological and Conceptual Challenges in ABMS:

- The methodological challenges in ABMS are critically examined in [66], [67], [68], and [69]. Especially relevant is [66], which elaborates on the core ideas and advantages of using ABMS for analysing complex adaptive systems. It highlights the rather unique capabilities of ABMS in this context.

- Work [67] highlights the flexibility of ABMS in research and teaching, discussing the specific challenges in model validation. This work underscores the applications of ABMS in academic settings.
- In [68], the integration of cooperative game theory in ABMS is explored, particularly in the complexities of human behaviour modelling. This integration showcases the potential benefits of ABMS in understanding intricate human dynamics.
- Lastly, [69] offers insights into agent-based control systems. It emphasizes key concepts underpinning agent-based computing, providing a deeper understanding of the foundational elements of ABMS and its suitability for complex systems engineering.

IV. Computational Tools and Aspects in ABMS:

- Works [70], [71] [72] [73] focused on the computational tools and aspects essential to ABMS. In particular, [70] emphasises the significance of computational tools in data management and visualization within ABMS, highlighting their role in enhancing analytical capabilities.
- Discussion of the popularity and application of various programming languages in ABMS is done in [71], with emphasis on the field's technological diversity. This exploration provides insight into the preferred ABMS tools by professional users.
- The advancement of computing research, including programming and parallel computing in ABMS, is examined in [72], which offers various examples and studies related to the evolving field of computational research on ABMS and its implications for future developments.
- Finally, [73] elaborates on the concept of ABMS and agent behavioural rules, discussing various software, toolkits, and methodologies. This discussion provides a comprehensive view of the different tools available in ABMS, along with an historical perspective.

4.2.2 Identified Knowledge Gaps and Current Trends in ABMS

A significant portion of current research in ABMS focuses on economic applications. There is a deficiency in SoS approaches, particularly within System Engineering Design. The research gap is particularly relevant in the field of Aircraft Conceptual Design. Here, COLOSSUS has the potential to contribute to the process of defining requirements and evaluating conceptual models, marking a significant innovation in this domain.
Additionally, COLOSSUS partners can extend their contribution to operational analysis in specialized sectors by using the ABMS toolkit for urban air mobility and wildfire combat cases. By studying these scenarios, it is possible to provide a comprehensive assessment of the impact of new concepts on existing practices. A key contribution of COLOSSUS and its framework could be the ability to identify emergent behaviours from detailed simulations. These behaviours have the potential to inform and inspire the development of new tactical approaches, particularly in the areas of urban air mobility and wildfire management.

4.2.2.1 Holistic SoS Engineering Methodologies

In the field of SoSE, there is still a notable gap in developing holistic methodologies. These methodologies should seamlessly integrate the various aspects of complex systems, as discussed in [56], [57], [58], and [59]. The core of SoSE, particularly in the context of ABMS, resides in the complex interaction of autonomous yet interconnected systems, with each system having its own operational and managerial characteristics, as defined in [56]. However, current literature and practices often consider these components in isolation. This approach results in a lack of a cohesive framework that could unify these disparate elements into a comprehensive whole. The empirical comparison in [57], contrasting agent-based with event-based modelling, highlights the necessity for a more integrated approach in characterizing SoS. Work [58] further reinforces this need by using the concept of Complex Adaptive Systems to understand network systems affected by disruptions. Despite these insights, a formalized, holistic methodology that encompasses the full spectrum of SoSE—from conceptualization to implementation—remains elusive. Furthermore, [59] explores the different applications of ABMS versus equation-based modelling, emphasizing the challenge of capturing the dynamic nature of SoSE within a single framework. This gap is significant not only academically but also in practical terms. It underscores the urgent need for a unified approach to effectively address the complex challenges posed by modern systems. There is significant potential for COLOSSUS to contribute to this lack of a unified approach. The extensive knowledge and diverse perspectives of the involved partners could lead to a breakthrough by proposing a unified, holistic framework for the first time.

4.2.2.2 SoS Architecting Processes

The domain of SoS architecting faces a significant gap in developing comprehensive processes for architecting, as indicated in [56] and [57]. There exists a wealth of knowledge on the description and theoretical aspects of SoS architectures. However, the field notably lacks a formalized process for the practical architecting of these complex systems. A parallelism is drawn in [56] between SoS development and Object-Oriented software development. It suggests that while theoretical concepts for describing SoS architectures are well-established, they are insufficient for guiding the actual architecting process. This limitation becomes more evident in [57], which discusses ABMS approaches. The focus of this discussion has been more on comparing modelling techniques, like agent-based and event-based approaches, rather than developing a structured process for SoS architecting. The absence of a comprehensive methodology for architecting SoS is not just an academic concern. It presents a significant challenge in practical applications, where exploring and realizing diverse SoS designs is crucial. This gap highlights the need for an innovative approach. Such an approach should draw from established architecture frameworks while being adaptable to the unique complexities and multifaceted nature of SoSs. Developing this process would be a major advancement in the field. It would enable practitioners to not only conceptualize but also effectively implement and manage SoS architectures across various domains.

4.2.2.3 Designing Behavioural Aspects in SoS

The design of behavioural aspects within SoSs is a critical yet underdeveloped area in SoS research. This gap is highlighted by the findings in [66], [62], [63], [59], and [69]. ABMS, as discussed in [66], presents a promising method for exploring complex dynamics. ABMS's bottom-up approach, simulating individual agent interactions to produce macroscopic phenomena, is particularly relevant. However, there is a noticeable lack of guidelines and frameworks for designing and exploring behavioural aspects in SoSs. This issue is exacerbated by the focus on domain-specific applications, such as mobility transitions in [62] and socio-political-economic predictions in [63], which lack a unified approach to behavioural design. In [59], it is discussed the necessity for distinct modelling approaches to accommodate the complexity and diversity of agent behaviours in SoSs. It suggests a shift from conventional equation-based modelling to more adaptable ABMS methods. Additionally, [69] highlights the potential of agent-oriented software engineering for conceptualizing purposeful agents within SoSs but acknowledges the lack of a systematic method for designing these interactions. The insights from these publications collectively underscore the need for a comprehensive methodology, something that can be tackled by COLOSSUS. This methodology should not only harness the strengths of ABMS but also provide a structured approach to designing and examining the collaborative and behavioural components of SoSs. Developing such a methodology would be instrumental in advancing our understanding and implementation of SoSs, enabling a more effective exploration of complex and interactive system behaviours.

4.2.2.4 Formal Methods in SoS Representations

The development of formal methods for representing SoSs poses a significant challenge, as highlighted in [56], [70], [63], [57], [59], [65], and [60]. [56] introduces the concept of SoS within the Object-Oriented Systems Approach, presenting a foundational framework. However, it lacks formal methodologies for a comprehensive representation of SoSs, particularly in collaborative and behavioural dimensions. Work [70] emphasizes the need for robust tools to manage high computational loads and Geographic Information System (GIS) data, underlining the complexity of accurately representing large-scale SoSs. The importance of modelling complex, nonlinear interactions between agents is stressed in [63], a crucial aspect that current formal methods do not fully capture. The comparative analysis in [57] between agent-based and event-based modelling approaches highlights the inadequacy of existing frameworks in capturing the dynamic interactions of independent agents in SoSs. Work

[59] contrasts traditional equation-based modelling with adaptable agent-based models, noting the suitability of the latter for transitioning from simulation models to adaptive control models in SoS contexts. In [65], it is discussed the proactive nature of agents in agent-based systems, indicating the need for better representation of individual agents and their interactions within SoSs. Similarly, [60] explores agent-based modelling as an alternative to variable-based techniques, emphasizing ABMS's potential to represent complex, interactive processes, a key element of SoSs. Collectively, these findings underscore the need to develop advanced formal methods and best practices. These should address SoS's unique characteristics like scale, complexity, and agent heterogeneity. Additionally, they should facilitate the emergence of large-scale behaviours through system interactions, potentially incorporating machine learning and reinforcement learning techniques. The COLOSSUS partners could utilize the project toolkit and their expertise in the aforementioned fields to provide solutions or contribute to reducing the knowledge gaps.

4.2.2.5 Challenges in Verification and Validation in ABMS for SoS Analyses

ABMS and SoS analyses face significant challenges in verification and validation, as evidenced by [67], [68], [57], [58], [59], and [69]. The complexity and scale of SoS analyses, especially when using ABMS, introduce unique challenges in ensuring simulation outcomes' reliability and accuracy. Work [67] highlights the difficulty of validating models due to the intricate nature of agent behaviours and system interactions. It mentions the need for advanced techniques to authenticate these complex dynamics accurately. Work [68] addresses the challenges stemming from the high degree of design freedom in ABMS. It notes that subtle changes in agent parameters can result in vastly different simulation outcomes, necessitating rigorous tuning and calibration. This challenge is further echoed in [57], which illustrates the complexities of comparing different modelling approaches, like agent-based and event-based models, in SoS characterization. Work [58] measures system resilience post-disruptions by making use of complex adaptive systems in network analysis and ABMS. It highlights emergent properties that require meticulous validation. The discussion in [59] takes the differences between ABMS and equation-based modelling. It points out that the adaptability of ABMS introduces additional layers of complexity in validation. Whereas [69] explores the agent-oriented approach in software engineering, underscoring the need for robust validation processes. These processes are essential to ensure that the autonomous and flexible behaviour of agents aligns with SoS objectives. These insights collectively shed light on the intricate challenges in verifying and validating large-scale SoS analyses through ABMS. They highlight the growing need for new and thorough methods that can precisely verify the detailed interactions of agents in complex systems.

4.3 Application and usage of SoS methodologies and processes

Due to the detailed and sensitive content needed for defining certain SoS architectures, finding documented real case-scenarios of implemented methodologies is often challenging, as companies could risk exposing important data. However, some papers provide a summarized yet comprehensive perspective on various industrial and research applications where SoSs have been designed and implemented. Jamshidi [74] provides a compiled study of different applications for SoSs in the industry as well as in research. Additionally, insightful examples of ABS applications are provided by Parunak [65] and Collins et al. [68], further enriching the understanding of SoS methodologies in practice.

5. Conclusion and outlook

This document has presented the content for Deliverable 5.1 of the COLOSSUS project. In summary, this deliverable explores the vital role of System-of-Systems Engineering (SoSE) in addressing complex challenges across engineering domains. The introduction defines System-of-Systems (SoS) as a collaborative network of constituent systems (CS) and introduces SoSE as a holistic methodology, emphasizing the need for a comprehensive approach to integration.

The document has served two primary objectives: Firstly, it has provided a State-Of-The-Art (SOTA) analysis for holistic SoSE and Agent-Based Modelling and Simulation (ABMS), drawing from COLOSSUS partners and external literature; and secondly, proposed a foundational frame of reference for defining SoS architecting processes within the project. The SOTA analysis identified current gaps in both SoSE and ABMS, where the most distinguishing one was the current lack of actual implementations of truly holistic SoSE methodologies. This together with a general lack of formal methods for representing SoSs in ABMS leaves a gap in how individual system behaviours that together create emergent SoS behaviours should be modelled and simulated.

This concise yet comprehensive exploration synthesizes internal and external insights, offering a valuable resource for the COLOSSUS project. Beyond highlighting the current state of knowledge, it also charts a possible course for future endeavours with SoSE for navigating the challenges posed by interconnected systems. Consequently, this deliverable serves as a key foundation upon which effective SoS solutions can be built within the COLOSSUS project.

6. References

[1] I. Staack, K. Amadori and C. Jouannet, “A Holistic Engineering Approach to Aeronautical Product Development,” *The Aeronautical Journal,* vol. 123, no. 1268, pp. 1545-1560, 2019.

[2] L. Knöös Franzén, A System of Systems View in Early Product Development: An Ontology-Based Approach, Linköping: Linköping Studies in Science and Technology, Dissertation No. 2317, Linköping University Electronic Press, 2023.

[3] A. Papageorgiou, Design Optimization of Unmanned Aerial Vehicles: A System of Systems Approach, Linköping: Linköping Studies in Science and Technology, Dissertation No. 2018, Linköping University Electronic Press, 2019.

[4] L. Knöös Franzén, An Ontological and Reasoning Approach to System of Systems, Linköping: Linköping Studies in Science and Technology. Licentiate Thesis No. 1907, Linköping University Electronic Press, 2021.

[5] S. Schön, Model Fidelity in Mission Scenario Simulations for System of Systems, Linköping: Linköping Studies in Science and Technology. Licentiate Thesis No. 1975, Linköping University Electronic Press, 2023.

[6] L. Knöös Franzén, I. Staack, P. Krus, C. Jouannet and K. Amadori, “A Breakdown of System of Systems Needs Using Architecture Frameworks, Ontologies and Description Logic Reasoning,” *Aerospace,* vol. 8, no. 118, 2021.

[7] D. A. DeLaurentis, K. Moolchandani and C. Guariniello, System of Systems Modelling and Analysis, New York: CRC Press Taylor & Francis Group, 2023.

[8] “Systems Engineering Guide for Systems of Systems,” Department of Defence, Washington, 2008.

[9] M. W. Maier, “Architecting Principles for System-of-Systems,” *Systems Engineering,* vol. 1, no. 4, pp. 251-313, 1998.

[10] “Mission Engineering Guide,” Department of Defense, Washington, 2020.

[11] P. Maupin, A.-L. Jousselme, C. Saurel and O. Poitou, “Situation Analysis and Performance Measurement: Application to Personnel Recovery,” in *13th International Conference on Information Fusion*, Edinburgh, UK, 2010.

[12] C. Saurel, O. Poitou and L. Cholvy, “Assessing the Usefulness of Information in the Context of Coalition Operations,” in *Information Quality in Information Fusion and Decision Making. Part of the Information Fusion and Data Science book series (IFDS)*, Springer, Cham, 2019, pp. 135-154.

[13] O. Poitou, P. Bieber, L. Simon and J. Ferreira, “Formal Architecture Modelling for Documenting and Assessing Aeronautics Maintenance: A Case Study,” ERTS, 2018.

[14] O. Poitou and C. Saurel, “Éviter l'échec des systèmes complexes: en construire collectivement une représentation formelle utile,” *Journées Francophones d’Ingénierie des Connaissances,* 2022.

[15] P. Bieber, F. Boniol, Y. Bouchebaba, J. Brunel, O. Poitou, T. Polacsek, L. Santinelli, N. Sesfelder and C. Pagetti, “A Model Based Certification Approach for Multi/Many-Core Embedded Systems,” ERTS, 2018.

[16] P. Bieber, F. Boniol, G. Durrieu, O. Poitou, T. Polacsek, V. Wiels and G. Martinez, “DGA TA: MIMOSA: Towards a Model Driven Certification Process,” ERTS, 2016.

[17] F. R. e. a. Golra, “Using Free Modeling as an Agile Method for Developing Domain Specific Modeling Languages,” in *In proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, ACM, 2016.

[18] F. R. Golra, A. Beugnard, F. Dagnat, S. Guerin and C. Guychard, “Addressing Modularity for Heterogeneous Multi-model Systems using Model Federation,” in *MODULARITY 2016: 15th International Conference on Modularity*, Malaga, Spain, 2016.

[19] O. Aïello, D. Sanchez Del Rio Kandel, J.-C. Chaudemar, O. Poitou and P. de Saqui-Sannes, “Populating MBSE models from MDAO analysis,” in *IEEE International Symposium on Systems Engineering (ISSE)*, 2021.

[20] O. Aïello, J.-C. Chaudemar and P. de Saqui-Sannes, “Sizing a Drone Battery by coupling MBSE and MDAO,” in *11th European Congress Embedded Real Time Systems (ERTS)*, 2022.

[21] A. A. Badra, O. Aïello and J.-C. Chaudemar, “Applying a Model-Based Systems Engineering Approach to Model an Unmanned Aerial Vehicle Mission,” in *IEEE International Systems Conference (SysCon)*, 2023.

[22] D. Amyot, S. Ghanavati, J. Horkoff and e. al., “Evaluating goal models within the goal-oriented requirement language,” *International Journal of Intelligent Systems,* vol. 25, no. AMYOT, Daniel, GHANAVATI, Sepideh, HORKOFF, Jennifer, et al. 8, pp. 841-877, 2010.

[23] P. Beery, T. Irwin, E. Paulo, A. Pollman, W. Porter and S. Gillespie, “Bridging Joint Operations and Engineering Management through an Operational Mission Architecture Framework,” *Engineering Management Journal (EMJ),* vol. 34, no. 3, 2021.

[24] T. Irwin, Operational Mission Architecture Framework: A Blended Architecture Methodology for Enabling Operational Capability, Dissertation - Defence Technical Information Center, 2018.

[25] S. Bouarfa, R. Aydogan and A. Sharpanskykh, “Formal modelling and verification of a multi-agent negotiation approach for airline operations control,” *Journal of Reliable Intelligent Environments,* vol. 7, 2021.

[26] J. Lee, M. Mitici, B. A. P. Henk, P. Bieber and F. Freeman, “Analyzing Emerging Challenges for Data-Driven Predictive Aircraft Maintenance Using Agent-Based Modeling and Hazard Identification,” *Aerospace,* vol. 10, no. 2, 2023.

[27] M. Ribero, J. Ellerbroek and J. Hoekstra, “Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning,” *Aerospace,* vol. 9, no. 9, 2022.

[28] M. Zoutendijk and M. Mitici, “Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem,” *Aerospace,* vol. 8, no. 6, 2021.

[29] V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske and S. F. Railsback, “The ODD protocol: A review and first update,” *Ecological Modelling,* vol. 221, no. 23, pp. 2760-2768, 2010.

[30] TransitWiki, “Four-Step Travel Model,” TransitWiki.org, 5 12 2019. [Online]. Available: https://www.transitwiki.org/TransitWiki/index.php/Four-step_travel_model. [Accessed 30 11 2023].

[31] “SUMO and Dynamic User Assignment,” German Aerospace Center (DLR), 10 11 2023. [Online]. Available: https://sumo.dlr.de/docs/Demand/Dynamic_User_Assignment.html. [Accessed 30 11 2023].

[32] “PTV VISUM - The world’s leading transport planning software,” PTV GROUP, 2023. [Online]. Available: https://www.ptvgroup.com/en/products/ptv-visum. [Accessed 30 11 2023].

[33] “TransCAD Transportation Planning Software,” Caliper Corporation, [Online]. Available: https://www.caliper.com/tcovu.htm. [Accessed 30 11 2023].

[34] “SUPPLEMENTARY GUIDANCE - Land Use/Transport Interaction Models,” Department of Transport, London, 2014.

[35] “MATSim Multi-Agent Transport Simulation,” MATSim Community - MATSim Association, 2023. [Online]. Available: https://www.matsim.org/. [Accessed 30 11 2023].

[36] “TransModeler Traffic Simulation Software,” Caliper Corporation, [Online]. Available: https://www.caliper.com/transmodeler/default.htm. [Accessed 30 11 2023].

[37] “Guide to Google Guice,” Baeldung, [Online]. Available: https://www.baeldung.com/guice. [Accessed 30 11 2023].

[38] O. Balci, “Verification, Validation and Accreditation of Simulation Models,” in *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, Georgia, 1997.

[39] S. M. Meentemeyer, B. Sauser and J. Boardman, “Analysing a system of systems characterisation to define system of systems engineering practices,” 2009.

[40] J. T. Boardman and B. J. Sauser, “System of Systems - the meaning of of,” *2006 IEEE/SMC International Conference on System of Systems Engineering,* pp. 6 pp.-, 2006.

[41] D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin and T. M. Shortell, Eds., Systems engineering handbook, Fourth Edition ed., Hoboken, N.J.: Wiley, 2015.

[42] *15288-2023 - ISO/IEC/IEEE International Standard - Systems and software engineering–System life cycle processes,* [Erscheinungsort nicht ermittelbar]: IEEE, 2023.

[43] J. P. T. Mo and R. C. Beckett, Engineering and Operations of System of Systems, Boca, Raton, 2018.

[44] “ISO/IEC/IEEE Systems and software engineering – Architecture description,” *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000),* pp. 1-46, 2011.

[45] M. Jamshidi, Ed., Systems of systems engineering, Boca Raton, Fla. [u.a.]: CRC Press, 2009.

[46] C. Keating, R. Rogers, R. Unal, D. Dryer, A. Sousa-Poza, R. Safford, W. Peterson and G. Rabadi, “System of systems engineering,” vol. 36, pp. 62-62, 2008.

[47] C. Keating and J. Padilla, “System of Systems Engineering Requirements: Challenges and Guidelines,” *Engineering Management Journal,* vol. 20, April 2015.

[48] J. Axelsson, “A systematic mapping of the research literature on system-of-systems engineering,” 2015.

[49] Z. Fang and D. DeLaurentis, “Multi-stakeholder Dynamic Planning of System of Systems Development and Evolution,” *Procedia Computer Science,* vol. 44, December 2015.

[50] C. E. DRIDI, Z. BENZADRI and F. BELALA, *System of Systems Modelling: Recent work Review and a Path Forward,* 2020.

[51] Z. Fang, “System-of-Systems Architecture Selection: A Survey of Issues, Methods, and Opportunities,” *IEEE Systems Journal,* vol. 16, pp. 4768-4779, September 2022.

[52] C. Guariniello and D. DeLaurentis, “Dependency Analysis of System-of-Systems Operational and Development Networks,” *Procedia Computer Science,* vol. 16, p. 265–274, December 2013.

[53] J. Dahmann, G. Rebovich, R. Lowry and K. Baldwin, “An Implementers' View of Systems Engineering for Systems of Systems,” *Aerospace and Electronic Systems Magazine, IEEE,* vol. 27, April 2011.

[54] P. Acheson, L. Pape, C. H. Dagli, N. Kilicay-Ergin, J. Columbi and K. Haris, “Understanding System of Systems Development Using an Agent- Based Wave Model,” *Procedia Computer Science,* vol. 12, p. 21–30, December 2012.

[55] J. Gideon, C. H. Dagli and A. K. Miller, “Taxonomy of systems-of-systems,” 2005.

[56] P. Acheson, C. Dagli and N. Kilicay-Ergin, “Model based systems engineering for system of systems using agent-based modeling,” *Procedia Computer Science,* vol. 16, p. 11–19, 2013.

[57] W. C. Baldwin, B. Sauser and R. Cloutier, “Simulation approaches for system of systems: Events-based versus agent based modeling,” *Procedia Computer Science,* vol. 44, p. 363–372, 2015.

[58] G. Pumpuni-Lenss, T. Blackburn and A. Garstenauer, “Resilience in complex systems: an agent-based approach,” *Systems Engineering,* vol. 20, p. 158–172, 2017.

[59] H. Van Dyke Parunak, R. Savit and R. L. Riolo, “Agent-based modeling vs. equation-based modeling: A case study and users’ guide,” in *International workshop on multi-agent systems and agent-based simulation*, 1998.

[60] E. R. Smith and F. R. Conrey, “Agent-based modeling: A new approach for theory building in social psychology,” *Personality and social psychology review,* vol. 11, p. 87–104, 2007.

[61] S. Abar, G. K. Theodoropoulos, P. Lemarinier and G. M. P. O’Hare, “Agent Based Modelling and Simulation tools: A review of the state-of-art software,” *Computer Science Review,* vol. 24, p. 13–33, 2017.

[62] M. Mehdizadeh, T. Nordfjaern and C. A. Klöckner, “A systematic review of the agent-based modelling/simulation paradigm in mobility transition,” *Technological Forecasting and Social Change,* vol. 184, p. 122011, 2022.

[63] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proceedings of the national academy of sciences,* vol. 99, p. 7280–7287, 2002.

[64] F. C. Billari, Agent-based computational modelling: applications in demography, social, economic and environmental sciences, Taylor & Francis, 2006.

[65] H. V. D. Parunak, “Practical and industrial applications of agent-based systems,” *Environmental Research Institute of Michigan (ERIM),* 1998.

[66] F. Klügl and A. L. C. Bazzan, “Agent-based modeling and simulation,” *Ai Magazine,* vol. 33, p. 29–29, 2012.

[67] K. J. Murphy, S. Ciuti and A. Kane, “An introduction to agent-based models as an accessible surrogate to field-based research and teaching,” *Ecology and evolution,* vol. 10, p. 12482–12498, 2020.

[68] A. J. Collins and S. Etemadidavan, “Interactive agent-based simulation for experimentation: a case study with cooperative game theory,” *Modelling,* vol. 2, p. 425–447, 2021.

[69] N. R. Jennings and S. Bussmann, “Agent-based control systems,” *IEEE control systems,* vol. 23, p. 61–74, 2003.

[70] A. Antelmi, G. Cordasco, G. D’Ambrosio, D. De Vinco and C. Spagnuolo, “Experimenting with Agent-Based Model Simulation Tools,” *Applied Sciences,* vol. 13, p. 13, 2022.

[71] K. Gunnarsson and O. Herber, *The Most Popular Programming Languages of GitHub's Trending Repositories,* 2020.

[72] D. Bouca, A (eds) 2010, Agent-Based Computing, Nova Science Publishers, Incorporated, Hauppauge. Ebook Central, 2024.

[73] C. M. Macal and M. J. North, “Agent-based modeling and simulation: ABMS examples,” in *2008 Winter Simulation Conference*, 2008.

[74] M. Jamshidi, “System of systems engineering - New challenges for the 21st century,” *IEEE Aerospace and Electronic Systems Magazine,* vol. 23, pp. 4-19, 2008.