Framework To Guide the Buildup of Simulation Capabilities for Heterogenous Urban Search and Rescue (USAR) Multi-Robot Simulation

by

Heng Huan Allan Law
B.S. Electrical Engineering (2012)

University of Illinois at Urbana-Champaign

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

MASTERS OF SCIENCE IN ENGINEERING AND MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Authored By: Heng Huan Allan Law
System Design and Management Program
January 28, 2023

Certified by: Eric Rebentisch, Ph.D.
Research Associate, Sociotechnical Systems Research Center & Lecturer,
Thesis Supervisor

Accepted by: Joan Rubin
Executive Director, System Design & Management Program

(This page left intentionally blank)

Framework To Guide the Buildup of Simulation Capabilities for Heterogenous Urban Search and Rescue (USAR) Multi-Robot Simulation

By
Heng Huan Allan Law

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM on JANUARY 28 2023
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF SCIENCE IN ENGINEERING AND MANAGEMENT

Abstract

This framework presents a systematic methodology of architecting a simulation setup that simulates a heterogenous multi-robot system which organizations can use to develop Robotics Urban Search and Rescue (USAR) Operations. The target audience would include system engineers or project managers who are looking to build up simulation capabilities to facilitate the development of heterogenous multi-robot systems.

The framework does this by systematically charting the basic architecture required in a Robotics USAR Operation and mapping these requirements on robot architectures. These robotic architectures will then be used as requirements for the simulation architecture needed to simulate these heterogenous multi-robot systems. The simulation architecture is then used to derive the changes needed in the organization to support such a capability build-up, as well as to be used to monitor the effects of different simulation requirements on the difficulty of building up the capability.

The framework applied to a representative scenario consists of USAR requirements drawn from previous disasters and the application of current robotics technologies found on the market. The resultant simulation framework will then be analyzed, and suggested courses of action to build up the simulation capabilities will be recommended.

Thesis Supervisor
Eric Rebentisch, Ph.D.
Research Associate, Sociotechnical Systems Research Center & Lecturer,
MIT System Design and Management

(This page left intentionally blank)

ACKNOWLEGDEMENTS

The work would not be possible without the guidance and support of my thesis supervisor, Dr Eric Rebentisch. I would like to thank Dr Rebentisch for the countless hours that he had dedicated in guiding me on the formulation of my thesis topic, and the valuable insights he shared with me about different digital engineering concepts and analysis tools.

I would also like to thank my wife for her moral support and understanding. Her words of encouragement provided me with deep motivation and brought me through the rough patches as I struggled to formulate my thoughts and ideas for my thesis topic.

(This page left intentionally blank)

Table of Contents

Table of Contents 7

List of Figures 10

List of Tables 11

1 Introduction and Motivation 12

1.1 Background 12

1.2 Objectives and Research Questions 12

1.3 Drivers Pushing for the Increased Use of Simulation 13

1.4 Motivation for Creating a Framework 13

1.5 Targeted Use-Case of Framework 14

1.6 Thesis Structure 14

2 Literature Research and Background Fundamentals 15

2.1 Introduction 15

2.2 Robot Search and Rescue Operations 15

2.2.1 Introduction 15

2.2.2 Types of SAR Operations 15

2.2.3 Opportunities of utilizing robotics 16

2.2.4 Key Tasks and Entities of General USAR Scenario 17

2.2.5 Robots Used in USAR Operations 18

2.2.6 Requirements of Robots used in USAR Tasks 19

2.3 Robotics Systems 19

2.3.1 General Requirements on Robotic Systems for USAR 19

2.3.2 Key Components of a Single Robotic System 20

2.4 Simulations 22

2.4.1 Different Levels of Simulations 22

2.4.2 Key Components in Robotic Simulators 23

2.5 Existing State of Robotics Simulation Frameworks 25

2.5.1 Examples of proposed simulation frameworks based on use-cases: 25

2.5.2 Examples of proposed simulation frameworks based on simulation fidelity. 25

2.5.3 Differences Between Available Frameworks and Proposed Framework 26

2.6 System Engineering Analysis Tools 26

2.6.1 Creating and Reading a Design Structure Matrix 26

2.6.2 Different types of DSMs 28
2.6.3 Tool selected for Analyzing DSMs 28
2.6.4 System Readiness Levels Derived from Integration Readiness Levels and Technology Readiness Levels 29
2.7 Summary 33
3 Framing High-Level Stakeholder-Related Needs for Simulations 34
3.1 Introduction 34
3.2 Framing Interaction Between Different Levels of Simulation 34
3.3 Describing Interaction Between Different Levels of USAR Robotic Simulation 35
3.4 Description of Motivation for Various Levels of Simulation 36
3.5 The focus of the Thesis Simulation Framework 39
3.6 Description of Hypothetical SAR Use-Case for Framework Application 39
3.7 Summary 40
4 Introduction to Framework 41
4.1 Introduction 41
4.2 Urban Search and Rescue Interaction 42
4.3 Multi-Robot System Interaction 44
4.4 Robotic Systems Interaction 45
4.5 Robotics System Simulation 47
4.6 Analysis of Simulation Architecture Using Engineering Tools 49
4.6.1 Building the Robot Simulation Architectures 49
4.6.2 Building the DSM 50
4.6.3 Populating the TRL and IRL Levels Within DSM 50
4.6.4 Calculating SRL 50
4.7 Conclusion 50
5 Application of Framework 51
5.1 Introduction 51
5.2 General Robot Types Chosen to Evaluate for Application of Framework 51
5.3 Different Strategies for Acquiring Robotic Systems 52
5.4 Selected Quadruped Robot 53
5.4.1 High-Level Hardware Architecture 54
5.4.2 Proposed Simulation Architecture 54
5.5 Selected Unmanned Aerial Vehicle 55

5.5.1 High-Level Hardware Architecture 56
5.5.2 Proposed Simulation Architecture 57
5.6 Selected Unmanned Ground Vehicle 58
5.6.1 High-Level Hardware Architecture 58
5.6.2 Proposed Simulation Architecture 59
5.7 Selected Multi-Robot Configuration 60
5.7.1 High-Level Hardware Architecture 61
5.7.2 Proposed Simulation Architecture 61
5.8 Choice of Simulation Engines and Software Middleware 62
5.9 Building the DSM for Analysis: 62
5.9.1 Building the DSM 62
5.9.2 Clustering the DSM. 63
5.9.3 Population of the TRL and IRL: 70
5.9.4 Calculating the SRL of the components 72
5.10 Conclusion 78
6 Discussion and Conclusion 79
6.1 Further Discussions on the Application of Framework 79
6.2 Limitations 80
6.3 Future Research 80
6.4 Conclusion 81
Bibliography 82
Appendix A 87

List of Figures

Figure 1. Department of Defense's Model Hierarchy for Defense-Related Simulations 22
Figure 2. Construction of DSM from a Directed Graph 27
Figure 3. Representation of interactions within the DSM 28
Figure 4. General Framework of Simulation Hierarchy 35
Figure 5. Interaction Between Various Levels of Simulation 36
Figure 6. Deriving Robotic Simulation Architecture of USAR Operation Needs 42
Figure 7. Block Diagram USAR Component Interaction 43
Figure 8. Block Diagram of Multi-Robot System Interactions 44
Figure 9. Diagram of Robot System Needs Interactions 46
Figure 10. Diagram of Robot Simulation Interactions 48
Figure 11. SPOT Robot Configuration Available from Boston Dynamic 53
Figure 12. High-level Hardware Architecture of Quadruped Robotic Platform 54
Figure 13. High-level block diagram representing main components of interest to simulate Quadruped Platform 55
Figure 14. Example of DJI M600 integrated with customized payloads 55
Figure 15. High-Level Hardware Architecture of UAV Platform 56
Figure 16. High-level block diagram representing main components of interest to simulate UAV Platform 57
Figure 17. Examples of robots with snake-like maneuverability 58
Figure 18. High-level Hardware Architecture of Ground Robotic Platform 59
Figure 19. High-level block diagram representing main components of interest to simulate Ground Platform 60
Figure 20. High-level Hardware Architecture of Multi-Robot System 61
Figure 21. High-level block diagram representing main components of interest to simulate Multi-Robot System 62
Figure 22. Overall DSM After Clustering 64
Figure 23. Clustering of Simulation Engine Components 65
Figure 24. Clustering of SPOT Robot Components 66
Figure 25. Clustering of UAV Robot Components 66
Figure 26. Clustering of UGV Robot Components 67
Figure 27. Clustering of Multi-Robot Simulation Components 67
Figure 28. Interactions Between Various Groups of Components Outside of Clusters 69
Figure 29. DSM with IRL, TRL and calculated SRL 73
Figure 30. Calculated System Readiness Levels of Components 74
Figure 31. Start state of SRL for UGV Simulation 76
Figure 32. Snapshot of present state of UGV Simulation 76
Figure 33. Simulation System Components with TRL and IRL Assignments 87
Figure 34. Quadruped Robot Simulation Components with TRL and IRL Assignments 88
Figure 35. UAV Robot Simulation Components with TRL and IRL Assignments 89
Figure 36. UGV Robot Simulation Components with TRL and IRL Assignments 90
Figure 37. Multi-Robot Simulation Components with TRL and IRL Assignments 91

List of Tables

Table 1. Kay Tasks for USAR Operation........17
Table 2. Main SAR Entities and Responsibilities........18
Table 3. How Robots were used in USAR Operations........18
Table 4. Robot System Components........20
Table 5. Proposed Simulation Activities Used Multi-Robot USAR Simulation........23
Table 6. TRL levels based on Clay et al. (2007) [55] and Interpreted Meaning for Thesis Framework........29
Table 7. Definitions and Criteria for Assigning Integration Readiness Levels to DSM. Based on Austin & York(2015) [53]........31
Table 8. Levels of Simulation, Stakeholders and Purpose........37
Table 9. Broad Needs of Example USAR Scenario........40
Table 10. USAR Operation Needs........43
Table 11. Multi-Robot System Addressing USAR Operation Needs........45
Table 12. Multi-Robot System Needs........45
Table 13. Individual Robotic System Addressing Multi-Robot System Needs........47
Table 14. Robotic System Needs........47
Table 15. Simulation Architecture Addressing Robotic System Needs........49
Table 16. Meeting Needs of Example USAR........52

1 Introduction and Motivation

1.1 Background

Major urban disasters such as the 1995 Oklahoma City bombing and the 1995 Kobe Earthquake led to the growing interest in deploying Search and Rescue (SAR) robots in urban disasters for Urban Search and Rescue (USAR) operations [1]. Since then, robots have been developed and deployed in various USAR operations requiring access to confined spaces inaccessible by traditional means to survey environments that are dangerous to the rescuers. Due to the complex nature of USAR operations, test and validation methods were needed to measure the performance of the robots, leading to the National Institute of Standards and Technologies (NIST) creating a physical arena to benchmark USAR robotic systems based on reference test arenas that simulated urban disaster scenarios [2].

The sole reliance on physical testing has its limitations; it could become costly and possibly dangerous to simulate a wide range of scenarios physically. DARPA initiated several challenges, including the DARPA Robotics Challenge (DRC) and the DARPA Subterranean (SubT) Challenge, which included a virtual challenge on top of the actual physical robot challenges. Simulations would be essential in developing more intelligent robotic technologies, as seen in the trend of autonomous vehicles.

There are several software simulation frameworks that are suitable for the simulation of specific robotic systems. Engineers would be able to select the most appropriate framework to meet their purpose. However, these simulation frameworks may not always meet the engineers' exact needs and choosing different simulations or robotic components will impact the readiness for simulation.

Simulating heterogeneous multi-robot systems presents a particular challenge due to the large number of components needed to be simulated, with each system potentially impacting the simulation setup in different ways. There is a lack of a managerial framework highlighting the implications of choosing various simulation components. Hence, this thesis attempts to provide a framework to allow simulation engineers or program managers to conceptualize a complex heterogeneous robot simulation setup, analyze the system readiness of the setup, and subsequently monitor the development of the simulation capabilities within the simulation setup.

1.2 Objectives and Research Questions

This thesis aims to address the following research objectives:

1. Organizations/stakeholders to frame the interactions of heterogenous multi-robot systems for USAR-type operations.

2. To evaluate how technical choices made will impact the readiness to implement the simulation setup.
3. To plan for and monitor the future buildup of capabilities based on requirements.

The following research questions will be answered throughout the thesis:

1. How do we systematically derive a multi-robot simulation architecture based on operational requirements?
2. How to we capture and model the resultant simulation architecture?
3. How do we represent the readiness levels of a chosen multi-robot simulation architecture based on technical decisions made?

The remaining sections within the introduction aim to briefly highlight the drivers that push for the increased use of simulation within the field of robotics development.

1.3 Drivers Pushing for the Increased Use of Simulation

As robots improve in their capabilities, they are beginning to be deployed in increasingly complex environments that require extensive testing for a successful development platform. One example is the push for autonomous vehicles, which requires extensive physical testing to develop, test, and validate the entire system. However, solely relying on physical testing of such robotic systems may not be practical for these various reasons; (1) the operational environment may not be feasible to replicate in a physical test site due to the scale or danger, (2) it will be costly in terms of time and money to carry out the repeated test. This is especially so when it comes to testing multiple robots.

Fortunately, simulation technologies are rapidly improving owing to an improvement in computing hardware, as well as improved simulation fidelity in areas, especially in physics simulation as well as visual simulation. This leads to the development of several simulations that specializes in running efficiently under several use cases, such as the simulation of high-fidelity physical simulations or high-fidelity visual simulations. To fully take advantage of such advancements in simulation technology, it is important for companies to have a strategy to build up their simulation capabilities according to various possible use cases.

1.4 Motivation for Creating a Framework

There are various simulation tools that are currently available in the industry for the use of simulating and developing robotics technologies [3], [4]. However, each of these tools has its own benefits and drawbacks and could be suited for a certain type of development over the other. The academic scene attempts to expand on these tools to form various robotic simulation frameworks by investigating various aspects of robotic development, such as communication, human-robot interaction, or multi-robot control algorithms. However, these simulation frameworks may not meet the specific needs of a robotic system and the successful implementation of a heterogeneous SAR Robot system would require a mixture of various simulation capabilities. It is not enough for a company to settle for a single tool without systematically evaluating the potential requirements of the simulation environment.

This thesis carries out a literature survey to investigate the various needs of SAR Robotics Systems and attempts to conceptualize a framework that will allow the reader to systematically determine the required simulation architecture for a given Robotic System of System configuration that would satisfy the needs of a specific task. Using this framework, the user can determine how the changing requirements impact the effort needed to build up this simulation capability.

1.5 Targeted Use-Case of Framework

The targeted audience of this framework is program managers in charge of building up simulation capabilities in their organizations. The simulation capability covered by this framework will be targeted at the design and development phase of both individual robots and multi-robot systems to be used in SAR operations.

1.6 Thesis Structure

The thesis Chapters are as follows:

1. Introduction
2. Literature review to introduce key components within USAR, Robotics SoS, and Simulations
3. Mapping stakeholder requirements from simulations based on open-source literature.
4. Introducing Basic Framework
5. Application of Framework with Specific Task, Robots, and Simulation Technologies
6. Discussion and Conclusion

2 Literature Research and Background Fundamentals

2.1 Introduction

Literature research will be used to drive the broad requirements of the various stages of a general conceptualization of a USAR robotic system architecture covered in Chapter 3. Three key aspects to be analyzed to achieve this process: Search and Rescue Operations, Robotics Systems, and finally, Robotic Simulation. This chapter goes on to introduce the current state of robotics simulation frameworks from literature and commercial use. Finally, the chapter will end with the introduction of key engineering tools that will be used to analyze the simulation architecture generated based on the requirements for USAR robotics development.

2.2 Robot Search and Rescue Operations

2.2.1 Introduction

This section introduces the various categories of SAR operations together with the unique challenges that each of them faces, with the focus being placed on the multiple tasks found within a USAR operation. A brief history of the robots used in various disasters, as well as the current efforts by organizations and research institutes in developing SAR robotics will also be discussed.

Queralta et al.(2020) [5]carried out an extensive review of multi-robot SAR from an algorithm's point of view. The following sub-section will use their definition of the three main groups of SAR operations to introduce the challenges and how they affect the requirements of robotic systems designed to operate in such environments.

2.2.2 Types of SAR Operations

<u>Maritime SAR:</u> Maritime Search and Rescue Operations involve the search and rescue of victims on the surface or subsurface of the water. These searches could involve potentially large swaths of areas [6] and could result in an expensive operation, especially if multiple assets are used to search these areas [7]. Possible disaster scenarios could include the search for several human victims, such as in cases of man overboard scenarios, to large-scale disasters that happens after natural disasters. [8] The challenges of subsurface search would involve potentially low visibility of the water and limited ability for communications, while surface operations would be subjected to obscured visibility due to waves [7]. With the aim of addressing these challenges, work has been done to develop multiple heterogeneous unmanned platforms. These include surface vehicles, underwater vehicles, as well as aerial vehicles that would work together to search and map the areas. [8]

Wilderness SAR: Wilderness SAR could encompass the search and rescue of victims in various different types of natural environments, including underground environments such as caves or mines [9], mountains [10], and forested areas [11]. The underground environment poses several challenges to the SAR personnel due to the possibility of toxic gases and dangerous terrain. Using robots within this environment is also challenging due to poor visibility, limited communications, and the lack of satellite positioning signals. For mountain-based SAR, the main challenge would be the harsh environment due to its altitude and temperature. Rescue operations will also need to be carried out quickly as survival rates of victims in such hash elements would severely degrade after 15mins. Lastly, the search for survivors in forested areas would be challenging as the victims could constantly be moving, hence complicating search strategies.

Urban SAR: Urban SAR operations cover natural or manmade disasters that happen in urbanized areas. Robots have been deployed in various situations that range from collapsed buildings due to earthquakes to flooding of towns or even fires []. Collapsed structures in these situations could pose challenges due to the dangers they pose to the search and rescue teams, while the trapped victims would be challenging to locate from the surface without being able to access the layers under the rubble. Other dangers to the SAR personnel could include the presence of hazardous materials within the structures, such as leaking gas pipes, rubble dust, or radioactive materials, as seen in the Fukushima disaster [1].

2.2.3 Opportunities of utilizing robotics

The brief survey of the various challenges posed by the three main types of SAR operations highlights various opportunities for the usage of robots to help meet these challenges. There are three main ways in which the use of robots can aid SAR operations. (1) The use of robots could lower the manpower needed to search large areas [12], hence potentially providing a faster response in various time-critical SAR operations. (2) These robots would be able to minimize the exposure of human life to the various dangers of the operational environment. (3) Robots could be able to search environments that may otherwise be inaccessible to humans or conventional equipment. The following information highlights the various types of robotic platforms that can be used for SAR operations.

Ground Robots: Able to carry out searches in areas that could be potentially dangerous to SAR personnel. Has the ability to search within confined areas that are hard to reach.

Areal Robots: Ability to cover the search and mapping of larger areas more efficiently as compared to ground or surface water robots. They can also survey areas that are difficult to access by ground.

Surface and Underwater Robots: Able to carry out maritime searches, such as searches underwater or out at sea.

2.2.4 Key Tasks and Entities of General USAR Scenario

This section highlights the key tasks that need to be carried out in a USAR operation, together with the key SAR entities that are involved or need to be considered in the USAR scenario. This will provide the basis for modeling interactions in Chapter 4.

Table 1 highlights the key tasks needed during a USAR operation. These key tasks, together with the SAR entities, can be used to formulate the general needs for a USAR operation found in Chapter 4.

Table 1. Kay Tasks for USAR Operation

Key Tasks	**Description**
Assessment of Disaster Site	One of the first few tasks to be done would be to carry out an assessment of the disaster sites. This would be critical in the planning of the search, as well as the safety of the SAR personnel. During this time, the structures will be assessed, while hazards that could affect the rescue personnel will be detected and identified [13] [14].
Situational Awareness	Throughout the operation, there would be a need for situational awareness through the surveillance of the disaster site [5]. Overall awareness would allow authorities better coordinate the disaster response, especially in the event of large disasters [15].
Interaction with Victims	Once the victims have been found, there would be a need to perform triage and possibly manipulate the victims to place them in safer areas [16]. The victims will then be extracted from the disaster site once the site is safe for SAR personnel to assess.
Ensuring the Safety of Rescuers	The USAR rescuers are exposed to several health and safety risks [17]. These risks include physical injuries, respiratory injuries, and diseases.
Quick Response	The ability to provide a quick response for the victims is crucial as the chances of survival of trapped victims quickly diminish over time [15].

Based on the USAR task force found on FEMA’s website, the main SAR entities can be categorized into six main groups that carry out the SAR operation, as found in Table 2.

Table 2. Main SAR Entities and Responsibilities

Entities	Description
Search Team	To search for and locate the victims in a disaster site.
Rescue Team	To extract the victims safely from the disaster site.
Site Assessment Team	The team will be responsible for assessing the safety of the structures and identifying possible hazards that could affect the SAR personnel on the ground.
Medical Team	The team will need to provide triage, as well as provide initial treatment for the injured victims.
Management Team	The team will need to provide overall coordination of the various SAR entities. They need to maintain situational awareness of the disaster site and reallocate resources when needed. This is especially so when there is a large-scale urban disaster caused by natural events such as earthquakes, hurricanes, flooding, etc. [14]
Logistics Team	To provide logistical support and equipment to the SAR operation.
Additional entities brought up in this section are as follows:	
Disaster Response Authorities	To be able to reallocate resources in the event of a large-scale disaster.
Victims	Two additional entities include the victims and the disaster site. The victims need to be located, extracted, and treated.
Disaster Site	The disaster site will present various challenges to the SAR response team and place requirements on how the team responds and what equipment can be used.

2.2.5 Robots Used in USAR Operations

The following table highlights several disasters in which robots were deployed in USAR operations. They represent the diverse ways that robots can be deployed in these operations.

Table 3. How Robots were used in USAR Operations

Event	Year	How Robots Were Deployed
World Trade Center Collapse	2001	Robots were used to search within the collapsed structures to locate the survivors as well as to identify potentially harmful gasses.
Hurricane Katrina Flooding	2005	Robots were used to enter damaged structures that could be unsafe for the SAR personnel.
Fukushima Nuclear Disaster	2011	Ground robots were used to enter the structures that contained radioactive materials. Underwater robots were deployed in the reactor coolant to monitor the condition of the reactors
Mexico Earthquake	2017	Articulated snake robots were used to enter collapsed structures that could not be assessed by other means.
Notre Dame Cathedral Fire	2019	UAVs were used to monitor the structures of the building, while firefighting robots were used to put out the fire.

2.2.6 Requirements of Robots used in USAR Tasks

Based on the disasters mentioned in Table 3, the following list contains some examples of the requirements of the robots used in USAR operations:

- Efficient Response, robots need to be portable and easily deployed by teams on the ground as the tasks are time critical. [1] [14].
- Ability to go places where conventional man-operated machines are unable to assess [14].
- Manipulate humans to reposition them safely and interact with the victims [16].
- Manipulate the disaster site to move away from dangerous obstacles [18].
- Building and maintaining communications and maps in areas of low connectivity, especially in areas with GNSS-denied environments [5].
- Multi-robot coordination to reduce the overall workload of the operator [5].
- HMI interaction development should enhance awareness of the operators controlling the robots, lower the cognitive load of various sensors, increasing the efficiency of the operator in controlling multiple robots [19].

These requirements can be used to drive the robot architectures of the simulated robots seen in Chapter 4, hence allowing the simulation setup to cater to various robot simulation configurations that meet these requirements.

2.3 Robotics Systems

The previous section briefly introduced the various types of robots used in USAR operations. This section aims to identify the general architecture of the robots and extract the key components that can then be referenced for simulation. With knowledge of the basic robot architecture, it then covers the key components within a heterogenous multi-robot system. Lastly, it will cover the various strategies of acquiring robot systems that can be used in the heterogenous multi-robot system.

2.3.1 General Requirements on Robotic Systems for USAR

In their review paper, Dorfetei et el [20] described the various possible requirements that developers should note when designing SAR robots to meet user needs. The nine needs highlighted in the paper can be separated into three broad categories: logistical, robot platform, and control needs. Based on the review paper and the literature review of the previous section, the points below highlight the various requirements of robotic systems.

Logistical Needs. The robotic system has a size and weight constraints as it is needed to be deployed in remote areas that do not allow for the use of heavy equipment for the unloading of the robots. Minimal manpower should be used to deploy and operate the robots to avoid diverting manpower away from the SAR teams.

Robot Platform Needs. Both the aerial and ground robots should ideally be able to operate during the day as well as at night in total darkness. During the operations, the robot should be durable enough to survive environmental conditions such as water and dust and should draw no more power than what can be provided onsite. Some sensing needs include the need for visual identification, detection, and geolocation of victims and the subsequent mapping of the disaster area. Microphones and speakers would also be needed to communicate with trapped victims.

Operator Control Needs. The robots should have stable connections to the operators despite the possibility of damage to the local communication infrastructure. Due to the high-stress environment of a USAR operation, the user-interface needs to be user-friendly.

Where possible, the simulation architecture derived from the framework needs to be able to support the development of multi-robot systems that meets the needs mentioned above.

2.3.2 Key Components of a Single Robotic System

Based on the robot architectures described in [18], [20]–[22], the key components that need to be considered within a single robotic system can be categorized into three main groups; Hardware Components, Software Components, and Human Operators. The following points list the various components and their purposes.

Table 4 highlights the subset of the various components that makes up a mobile robotic system.

Table 4. Robot System Components

Type of Component	**Components**	**Description**
Structural	Mechanical Chassis	The chassis forms the structural frame of the robot on which all the other components will be mounted.
Actuation	Motors, Wheels, Tracks, Manipulation Arms, Propellers	These components can be broadly categorized into two broad uses. The first is for propulsion to move the robot through its environment, and the second is for manipulating its environment.

Sensors	Electro-Optical/Infra-Red, LiDAR, microphones, gas sensors	These components are responsible for sensing the disaster environment, which includes structures as well as victims.
Computing	Higher Level Computing, Lower Level Computing	More complex robots would often require various levels of computing to handle various aspects of the robot. The lower-level computing would handle the basic robot functionalities, which include controlling actuators and intelligence for low-level platform controls such as robot movement. High-level computing could involve a higher level of robot autonomy, such as robot perception, environmental mapping, and robot coordination.
Communication	Wireless Modules, Network Modules.	The communication modules allow for the sending of communication between various subcomponents on the robot or between different robots and ground control stations.
Power	Batteries, Tethered Power Delivery	The power components provide the energy needed for the various electronic components.
Operator Control	Ground Control Station (GCS), Operator	For a single robot system, the GCS will allow an operator to have control of a single robot system, while the GCS used for a multi-robot system will allow the operator to have one-to-many control of robots that are connected to the GCS.

These key components would have various interactions and possible dependencies that could be modeled in the final multi-robot simulation architecture that will be developed in chapter 4.

2.4 Simulations

The first part of this section will cover the various levels of simulations that can be carried out during the development cycle of a multi-robot system. The main types of simulations that can be carried out for robotic development are generalized based on the similarities between the various types of simulation used in developing military technology. Once the targeted levels of simulation for the robotic development have been identified, key simulation components can then be identified. This section ends with examples of various different robot simulators that could be used.

2.4.1 Different Levels of Simulations

Simulation brings several benefits to the development of complex multi-robot systems. Some targeted objectives include accelerating and lowering the cost of development, reducing the risk and hazards of physical testing by reducing the amount of testing, and providing large amounts for the development of algorithms in a cost-effective way [23], [24]. Loper and Turnitsa (2012) [25] used the description of Live, Virtual and Constructive (LVC) simulation to frame the various simulation techniques used in military combat simulations. Live Simulation involves real people using operating real equipment, while Virtual Simulation involves a mix of virtual and live components. Constructive Simulation involves fully simulated entities. This thesis will focus on the use on the use of Virtual Simulations, in which there is a mix of simulated entities and real entities that needs to be developed such as software algorithms and hardware components.

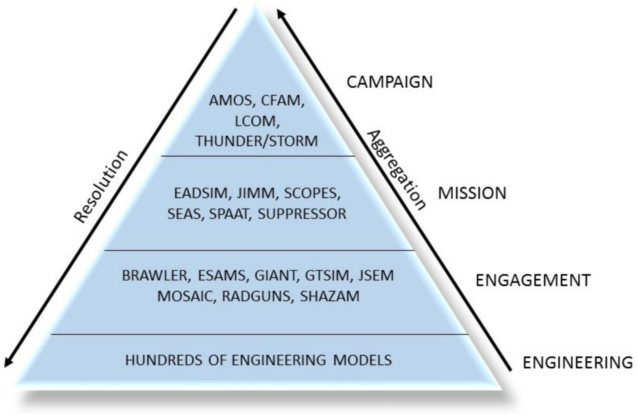


Figure 1. Department of Defense's Model Hierarchy for Defense-Related Simulations

Hill et al. (2018) [26] introduced a pyramid shown in Figure 1 that highlights how some of the US Airforce models used in constructive simulations contribute to the simulations done at various operational levels.

The left arrow represents the resolution of the models increases as the scope of the simulation decreases, while the right side of the pyramid represents the aggregation of these models to cover a wider scope of operational simulation. Although Hill et al. (2018) introduced the pyramid to describe constructive simulation, the concept was adapted in this thesis to describe the various simulation efforts that can be used to develop multi-robot operations for USAR activities as shown in Table 5.

Table 5. Proposed Simulation Activities Used Multi-Robot USAR Simulation

Level	Type of Simulation	Targeted Purpose of Simulation	Examples found in Sources
1	USAR Concept Of Operations Simulation	Determine resources needed to fulfill the USAR objectives. This is similar to the campaign simulation outlined in Figure 1.	[27], [28]
2	Multi-robot Mission Modeling and Analysis Simulation	Determine configuration and performance goals of the multi-robot system needed to meet the mission objectives required by the USAR operations.	[29]
3	Multi-robot system-level simulation	Development of a multi-robot system based on the performance goals required.	[17], [30], [31]
4	Robot system-level simulation	Development of individual robot systems to meet the larger system needs.	[18], [32]
5	Robot Sub-Component Simulation	Development of robot sub-components to meet robot system requirements. Examples of subcomponent simulation would include structural, electronics, or electromagnetics simulations.	[33], [34]

The simulation architecture derived from this framework will be used by engineers developing multi-robot systems. Simulation levels 1 and 2 will not be covered by this framework as they are targeted at decision-makers who are evaluating the needs or effectiveness of the robotic system for USAR Operations. Simulation levels 5 will not be covered in this simulation framework as the type of simulations used for sub-component simulation is not covered by robotic simulators. Hence, only simulation levels 3 and 4 will be outlined by this framework.

2.4.2 Key Components in Robotic Simulators

Collins et al. [3] summarized a range of different robot simulators that are targeted at different robotic applications, each with its own strengths and shortcomings. However, the components within these engines can be generalized into the following key components: The Simulation engines, software and hardware interfaces, virtual models, and human interaction interfaces.

Simulation Engines. Robot simulators typically contain one or more simulation engines that handle two main types of simulations. The first would be the mechanical or physical simulation of components found within the simulation world. These include the simulation of forces exerted by the robot's actuators or contact interaction between physical components within the simulation environment [3]. Examples of physics simulation engines include Open Dynamics Engine (ODE), pyBullet, and MuJoCo Simulation. The second would be the rendering engine that is responsible for generating images based on the visual properties of the entities found within the simulation environment [35]. Examples of such engines include OGRE, OpenGL, and Blender. The other simulation that is not widely covered in popular simulation engines is the simulation of the communication network and its effects. Various papers proposed integrating existing robotic simulators with network simulators to simulate the effects of imperfect wireless communication based on the position of the robot or the obstacles between the robots [36]–[39].

Virtual Models. These models consist of entities that need to be represented or simulated in the virtual environment. Some examples include sensor models and actuator models. Mechanical models and USAR environment models. [40], [41]. The USAR environment models represent the virtual entities simulated for the purpose of the robot interaction. Examples include the terrain, buildings, and the victims. The sensor models are responsible for how the robot senses the environment and the information available to the operator or the robot algorithms to make decisions. The robot is able to move around and manipulate its environment through the use of its actuator models and mechanical models.

Software Interfaces. Robot simulations are used in the development of various components within the robot. Examples range from the development of robot software or algorithm when Software-in-the-loop is used to the development of robot system hardware when the hardware in the loop is used [35]. Robotic middleware is used as the software interface between the components being developed and the simulated inputs/outputs [42]. This allows the code or hardware to be used in the actual platform with minimal changes.

Hardware Interfaces. Hardware interfaces to the computing solution that runs the robot simulator can be used for hardware-in-the-loop purposes [43] or to interface hardware to be used for human-robot-interaction. A single common interface such as Wi-Fi or USB that carries all the simulated inputs and outputs can be used between the computing hardware and the simulator [44], [45], hence requiring no specialized hardware for the interface. Another method involves simulating individual hardware digital or analog interfaces produced by the sensors and actuators, requiring specialized equipment to produce such signals [32].

Human-Robot Interaction. Human-Robot Interactions (HRI) consist of human operators interacting with robotic systems through the use of a computer and a graphical interface. These interactions can consist of one-to-one interactions in which an operator controls a single robot [19], [41] or one-to-many interactions in which a human operator has the ability to control a swarm of robots [46].

Simulation Scenario and Logic. The simulation scenarios or simulation logic controls various virtual entities and their behaviors within the simulation environment. For example, the CARLA simulator has the capability of simulating complex traffic and pedestrian behavior in crowded urban scenarios, hence allowing for the testing of Autonomous Vehicles' behavior [47].

Graphical User Interface. The graphical user interface represents the human-computer interface in which the user of the simulator is able to generate new scenarios or monitor the ongoing progress of the simulation from a third-person perspective [48].

2.5 Existing State of Robotics Simulation Frameworks

This section highlights robotic simulation frameworks proposed by various research groups developing robotic applications. The frameworks can be categorized into two main types; frameworks that focus on the use case and frameworks that focus on accurate simulation fidelity. All these robotic frameworks consist of functional simulation software that can be used for their application.

2.5.1 Examples of proposed simulation frameworks based on use-cases:

Advanced Framework for Simulation, Integration, and Modeling (AFSIM) is a simulation framework targeted at defense-related mission and engagement level simulation [48]. This framework provides the U.S. (?) Air Force and its collaborators with a common set of tools to model and simulate advance technological concepts, hence reducing the overhead of creating specialized simulation tools while improving the ability to compare different technology concepts provided by different vendors.

Bottger et al. (2019) described a medical robotics simulation framework to investigate optimal kinematic structures and configurations of surgical robotic arms that would be used to carry out operations on patients [49]. This framework consists of simulation software, patient models as well as evaluation metrics that can be used for surgical robot arm simulations.

2.5.2 Examples of proposed simulation frameworks based on simulation fidelity.

Kudelski et al. (2012) described a framework for the realistic simulation of networked multi-robot systems [36]. This framework provides higher fidelity networked communications simulation by integrating advanced network simulation tools together with a multi-robot simulator. The use of this framework demonstrated how realistic network simulation influenced the behaviors and performance of simulated multi-robot systems.

Zhu et al. described a modular simulation framework called Robosuite that can be used to benchmark robot learning algorithms [50]. The framework makes use of the MuJoCo physic engine that provides high-fidelity mechanical simulation while also providing the ability to perform benchmarks between different robot learning algorithms.

2.5.3 Differences Between Available Frameworks and Proposed Framework

The various robotic simulation frameworks that were found through the literature survey consisted of functional engineering simulation tools that engineers can utilize while developing various algorithms for specific use-cases of a robotic/multi-robot system. However, these simulation frameworks may not always be immediately usable as the multi-robot system may have different system needs that may require the combination of several different simulation frameworks, such as the example highlighted by Kudelski et al. (2012) [36]. The functional robotic simulation framework fails to capture how different simulation architectures and operational needs affect the readiness of an organization's simulation capabilities. In contrast, the simulation framework described by this thesis is targeted at engineering managers with the aim of providing them with a structure in which they are able to formulate simulation architectures and subsequently analyze and manage the capability development of the simulation architectures.

2.6 System Engineering Analysis Tools

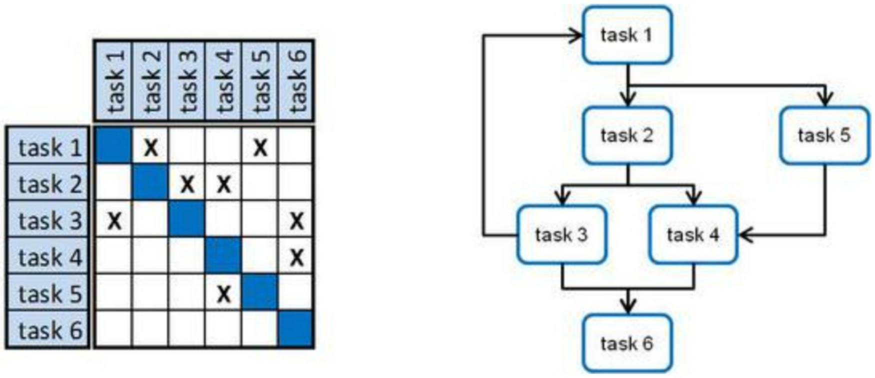
This section highlights the various engineering tools that can be used to formulate and analyze the simulation architecture. The first tool helps with the architecting of a multi-robot simulation setup by viewing the multi-robot simulation from a system-of-systems perspective. Once the architecture has been done, the Design Structure Matrix (DSM) can then be used to capture the interactions between the simulation components. With the aim of gaining insights into the readiness of the architected simulation system, Integration Readiness Levels (IRL) and Technology Readiness Levels (TRL) can be used to formulate the System Readiness Levels (SRL) of each component in the system.

2.6.1 Creating and Reading a Design Structure Matrix

In the book "Design Structure Matrix Methods and Applications" [51], Steven D. Eppinger and Tyson R. Browning provided an overview of how DSMs can be constructed and analyzed, together with various industrial applications of the DSM. The information found in this subsection is extracted from that publication.

In the book, Eppinger and Browning described the DSM as a network modeling tool that can be used to represent the interactions between various components within a system architecture. The DSM could represent large graphs that would otherwise be difficult to represent using traditional graphical representations, hence allowing the user to carry out further analysis with the architecture, such as re-clustering or sequencing, or other numerical analysis if the interconnectivity of the components is weighted.

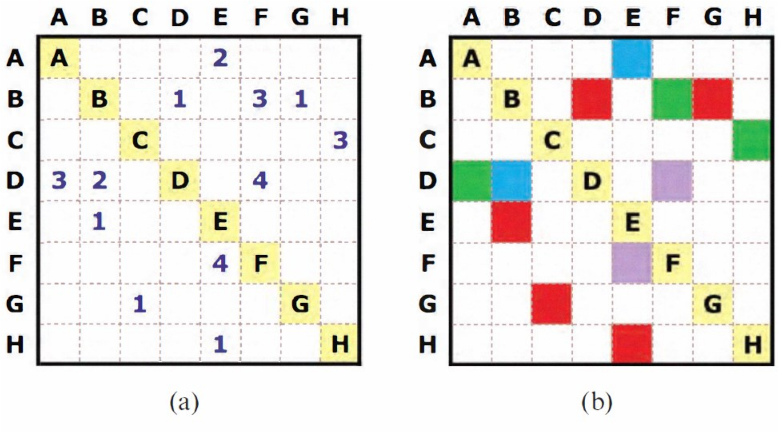
A DSM contains a square matrix of N x N matrix of a set of N elements which represents the mapping of interactions between the components. There can be two different ways to represent the directionality of the graph. This thesis will adopt the same convention as the book, in which the columns will represent the input rows, as shown in Figure 2.



	task 1	task 2	task 3	task 4	task 5	task 6
task 1	■	X			X	
task 2		■	X	X		
task 3	X		■			X
task 4				■		X
task 5				X	■	
task 6						■

Figure 2. Construction of DSM from a Directed Graph (image source: [51])

The values found within the cells of a DSM could contain numbers that describe various numerical properties of the interactions between the components, such as the strength of interactions, as shown in Figure 3a. It can also contain colors or other forms of labelling to represent the type of interaction, as shown in Figure 3b.



(a)

	A	B	C	D	E	F	G	H
A	A				2			
B		B		1		3	1	
C			C					3
D	3	2		D		4		
E		1			E			
F					4	F		
G			1				G	
H					1			H

(b)

	A	B	C	D	E	F	G	H
A	A				🟦			
B		B		🟥		🟩	🟥	
C			C					🟩
D	🟩	🟦		D		🟪		
E		🟥			E			
F					🟪	F		
G			🟥				G	
H					🟥			H

Figure 3. Representation of interactions within the DSM (image source: [51])

2.6.2 Different types of DSMs

With reference to the book, Eppinger and Browning (2012) [51] also described three different types of DSMs. The first is the Product Architecture DSM which represents the interaction of various elements within a complex product or system. The second is the Organization Architecture DSM, which represents the network of people that forms the structure of business units or organizations. The third is the Process Architecture DSM, which represents various activities and their interactions within an organization or projects. A Multidomain Architecture DSM could be generated by combining some or all three of the different DSMs.

The simulation architecture derived from the proposed framework will represent a complex simulation system required to simulate and develop a heterogenous multi-robot system. Hence, the Product Architecture DSM will be used to capture the interaction between the different simulation components.

2.6.3 Tool selected for Analyzing DSMs

The DSM analysis tool selected for use in this thesis is a research tool called the Cambridge Advanced Modeler (CAM) 2 that can be used without any fees [52]. CAM2 provides four main tools: Process Modeling, Dependency Modeling, Change Modeling, and Visual Analytics. The main tool that was selected to be used in this thesis is the Dependency Modeling tool, in which the clustering algorithm was used to explore how components can be better grouped within the simulation architecture.

2.6.4 System Readiness Levels Derived from Integration Readiness Levels and Technology Readiness Levels

Austin and York (2015) [53] described the need to adopt system-level metrics that place greater emphasis on the integration between various components. This will allow technical and managerial decision-makers to make better-informed decisions throughout the product life cycle. The paper describes how System Readiness Assessment could be carried out. For this thesis, the System Readiness Level (SRL) metrics will be adopted. The following sections describe how the SRL can be calculated from the Technology Readiness Levels (TRL) and Integration Readiness Levels (IRL).

2.6.4.1 Technology Readiness Level Metric

The Technology Readiness Level (TRL) is a metric that was developed by NASA to assess and represent the maturity of technology [54] and subsequently adopted by the DoD to assess the maturity during the procurement of projects. However, the definitions of TRL used are mainly aimed at hardware [55]. Hence, Clay et al. (2007) [55] adapted the standard TRL definitions with the aim of describing the capability maturity of the various components found in modeling and simulation development. The adaptation was needed as the modeling and simulation capabilities are a combination of software, hardware, and expertise, as opposed to being a physical device. This thesis will adopt similar definitions. The definition of the capability maturity described by Clay et al. (2007) [55], together with what it means for the framework described in this thesis, is shown in the table below.

Table 6. TRL levels based on Clay et al. (2007) [55] and Interpreted Meaning for Thesis Framework

TRL	Capability Maturity	What it means for COTs component in Simulation Framework
1	Concept Phase: Basic principles identified.	The basic functionality of the component has been identified
2	Concept Phase: Technology concept and/or app formulated.	Key functional requirements of the component have been formulated.
3	Concept Phase: Proof of concept initiated.	The component has been tested for basic functionality
4	Prototype Phase: Concept demonstrated on a 'toy'/lab problem.	The component has been tested on a toy problem scenario
5	Prototype Phase: Key elements demonstrated on a realistic problem.	The component has been tested on a realistic simulation scenario
6	Prototype Phase: System model demonstrated a realistic problem.	The component has been integrated into the wider simulation system and tested on a realistic simulation scenario
7	Production Phase: System demonstrated on a realistic problem in production.	The integrated component has been tested in a real-world scenario
8	Production Phase: System completed and qualified on production through test and demonstration.	Integrated Component has been tested on a real-world scenario and qualified with actual test data
9	Production Phase: System completed and in ongoing production use.	Integrated Component has been tested on a real-world scenario and qualified with actual test data

2.6.4.2 *Integration Readiness Level Metric*

The IRL metric will be used to represent the maturity of the interfaces between two or more components [53]. This is an important metric to describe the overall readiness of the system, as the use of TRL alone does not capture the readiness of the interfaces between the various modules. Hence, the TRL can be higher than the IRL. For the assignment of IRL in this thesis, it is assumed that the IRL cannot be higher than the assigned TRL of the component, as the component needs to be at sufficient maturity before it can be integrated successfully.

Table 7 will be used to assign the IRL between components.

Table 7. Definitions and Criteria for Assigning Integration Readiness Levels to DSM. Based on Austin & York(2015) [53]

IRL	Definition	Evidence Description
0	No Integration	No integration between specified components has been planned or intended
1	A high-level concept for integration has been identified.	**Principal integration technologies have been identified.** Top-level functional architecture and interface points have been defined High-level concept of operations and principal use cased has been started
2	There is some level of specificity of requirements to characterize the interaction between components	**Inputs/outputs for principal integration technologies/mediums are known, characterized and documented.** Principal interface requirements and/or specifications for integration technologies have been defined/drafted
3	The detailed integration design has been defined to include all interface details.	**Detailed interface design has been documented.** System interface diagrams have been completed Inventory of external interfaces is completed and data engineering units are identified and documented
4	Validation of interrelated functions between integrating components in a laboratory environment.	**Functionality of integrating technologies (modules/ functions/ assemblies) has been successfully demonstrated in a laboratory/ synthetic environment.** Data transport method(s) and specifications have been defined
5	Validation of interrelated functions between integrating components in a relevant environment.	**Individual modules tested to verify that the module components (functions) work together.** External interfaces are well defined (e.g., source, data formats, structure, content, method of support, etc.)
6	Validation of interrelated functions between integrating components in a relevant end-to-end environment.	**End-to-end Functionality of Systems Integration has been validated.** Data transmission tests completed successfully
7	System prototype integration demonstration in an operational high-fidelity environment.	**Fully integrated prototype has been successfully demonstrated in actual or simulated operational environment.** Each system/software interface tested individually under stressed and anomalous conditions Interface, Data, and Functional Verification complete
8	System integration completed and mission qualified through test and demonstration in an operational environment.	**Fully integrated system able to meet overall mission requirements in an operational environment.** System interfaces qualified and functioning correctly in an operational environment
9	System Integration is proven through successful mission-proven operations capabilities.	**Fully integrated system has demonstrated operational effectiveness and suitability in its intended or a representative operational environment.** Integration performance has been fully characterized and is consistent with user requirement

2.6.4.3 System Readiness Level Metric

Austin and York (2015) [53][1] introduced three types of SRL; Component SRL, Composite SRL, and SRL. The Component SRL is the readiness level of an individual component of the system and its interfaces, while the Composite SRL represents the readiness level of the entire system. Both levels are normalized from 0 to 1. The SRL represents the Composite SRL when scaled from 1 to 9, hence allowing similar comparisons to IRL and TRL metrics. This thesis will focus on the calculation of and interpretation of the SRL metrics as it will allow for greater resolution in the analysis of the readiness levels of various components.

For a system with n components, the TRL and IRL matrix can be represented as follows:

$$[\mathrm{TRL}]_{nx1} = \begin{pmatrix} TRL_1 \\ TRL_2 \\ \ldots\ldots \\ \ldots\ldots \\ TRL_n \end{pmatrix} \quad (1) \qquad [\mathrm{IRL}]_{nxn} = \begin{pmatrix} IRL_{11} & IRL_{12} & \ldots & \ldots & IRL_{1n} \\ IRL_{21} & IRL_{22} & \ldots & \ldots & IRL_{2n} \\ \ldots & \ldots & IRL_{33} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ IRL_{n1} & IRL_{n2} & IRL_{n3} & \ldots & IRL_{nn} \end{pmatrix} \quad (2)$$

The values of the TRL and IRL must be normalized by the highest levels of the TRL and IRL scale, which is the value of 9. Hence, the values will range from 0 to 1. The diagonals of the IRL will have a value of 1, which represents the TRL of the component, without any interactions, after the matrix multiplication has been carried out in equation (4).

The interfaces of the system components described by Austin and York (2015) [53] form an undirected graph. Hence the resultant IRL matrix will be symmetric. The resultant Component SRL can be calculated as follows.

$$[\mathrm{SRL}]_{nx1} = [\mathrm{IRL}]_{nxn} \times [\mathrm{TRL}]_{nx1} \quad (3)$$

$$\begin{pmatrix} SRL_1 \\ SRL_2 \\ SRL_3 \\ \ldots \\ SRL_{10} \end{pmatrix} = \begin{pmatrix} IRL_{1,1} & IRL_{1,2} & \ldots & \ldots & IRL_{1,10} \\ IRL_{2,1} & IRL_{2,2} & \ldots & \ldots & IRL_{2,10} \\ \ldots & \ldots & IRL_{3,3} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ IRL_{10,1} & IRL_{10,2} & IRL_{10,3} & \ldots & IRL_{10,10} \end{pmatrix} X \begin{pmatrix} TRL_1 \\ TRL_2 \\ TRL_3 \\ \ldots \\ TRL_{10} \end{pmatrix} \quad (4)$$

---

[1] Equations 1 to 5, and equation 7 are obtained from Austin and York (2015) [53] without any modifications

$$Component\ SRL_i = \frac{SRL_i}{m_i} \quad (5)$$

The component SRL can be calculated by multiplying the IRL by the TRL as shown in (3) and (4) and subsequently normalized by all the interfaces as shown in (5), where m_i equals the number of interfaces with the addition of the component itself. An example of this calculation can be found in the paper presented by Austin and York (2015) [53].

However, this thesis chose to represent the interaction of the simulation components with greater resolution; hence the interactions will be directional. This allows for the capture of the effort needed for an informational flow between a bi-directional interface as compared to a unidirectional interface. The application of the formula above would only consider the inputs into the component and ignore the outputs. Hence, the above calculation needs to be tweaked as follows:

$$[\mathrm{SRL}]_{n\times 1} = [\mathrm{IRL}]_{n\times n} \times [\mathrm{TRL}]_{n\times 1} + [\mathrm{IRL}]^T_{nxn} \times [\mathrm{TRL}]_{n\times 1} \quad (6)$$

$$Component\ SRL_i = \frac{SRL_i}{m_i} \quad (7)$$

The SRL can now be calculated by considering the inputs and the outputs by transposing the IRL matrix. The Component SRL can then be normalized across all the directional interfaces in equation (7), where m_i equals the total number of directional interfaces with the addition of 2 of its own interactions.

2.7 Summary

This chapter briefly introduces the three main areas of interest that will be used to formulate the framework: USAR Operations, Robotic Systems, and Simulation Systems. The next step is incrementally formulating the final multi-robot simulation architecture based on the USAR operational needs. Unfortunately, the final multi-robot simulation architecture may not be immediately compatible with current readily available functional robotics simulation frameworks, as highlighted in subsection 2.4. Hence, decision-makers such as system engineers or project managers need to understand how their choice of robotic systems and simulation elements impacts the readiness of their simulation capabilities, as this could impact the allocation of resources needed to build such capabilities. Describing the complex simulation architecture as a DSM and applying the SRL will allow managers to appreciate better the system readiness levels of their required simulation architecture.

3 Framing High-Level Stakeholder-Related Needs for Simulations

3.1 Introduction

In Section 3.2, the various levels of simulations were introduced. This chapter aims to frame the interactions between these various levels of simulations and to highlight the layers of simulations that the framework in this thesis aims to address. The simulation setup addressed by this framework is only designed to target specific layers of simulation; hence it is important to find out how the various stakeholders will benefit from such simulations. The rough needs from these stakeholders could then be used to drive the simulation requirements and, subsequently, the simulation architectures found in this framework.

Generic interactions between different levels of simulation are provided, as shown in Figure 4. It is then applied to the various levels of simulation covered in Section 3.2 to form a complete diagram that highlights how the various levels of simulation benefit from each other, as shown in Figure 5. The possible stakeholder needs based on literature research are also tabulated in Table 8. Lastly, the levels of simulation of interests are then identified, driving the simulation requirements and architecture.

This chapter ends by briefly introducing the chosen USAR scenario that will be used as an example to apply the framework that will be described in the next chapter.

3.2 Framing Interaction Between Different Levels of Simulation

Borrowing the concept of the DoD’s Model Hierarchy for defense-related simulation, the general framing of the interaction between the various levels of simulation can be described in Figure 4 below. Similar to the DoD’s Model Hierarchy, the higher-level simulation levels represent a higher level of abstraction and aggregation, and lower levels of simulation represent the higher resolution of simulations in which higher fidelity simulations are carried out. The higher-level simulation can be done to conceptualize the requirements needed for the system, and the specifications of the subcomponents of the system can then be propagated down to the lower-level simulations. The lower-level simulation of the subcomponents would be higher fidelity in nature, and this would allow for the investigation of whether the subcomponents would be able to meet those specifications. The lower-level simulation would be able to feed the results back to the higher-level simulation through reduced-order models, which would have greater accuracy based on the specifications that it was given. Hence, this forms an iterative loop in which higher-level simulation drive sub-system needs, while lower-level simulations would provide better clarity to the feasibility of such needs and the actual expected performance. Compared to DoD’s Model

Hierarchy, Figure 4 explicitly highlights the flow of information from one level to another, as well as the possibility of iteratively refining the developing and building on the simulations.

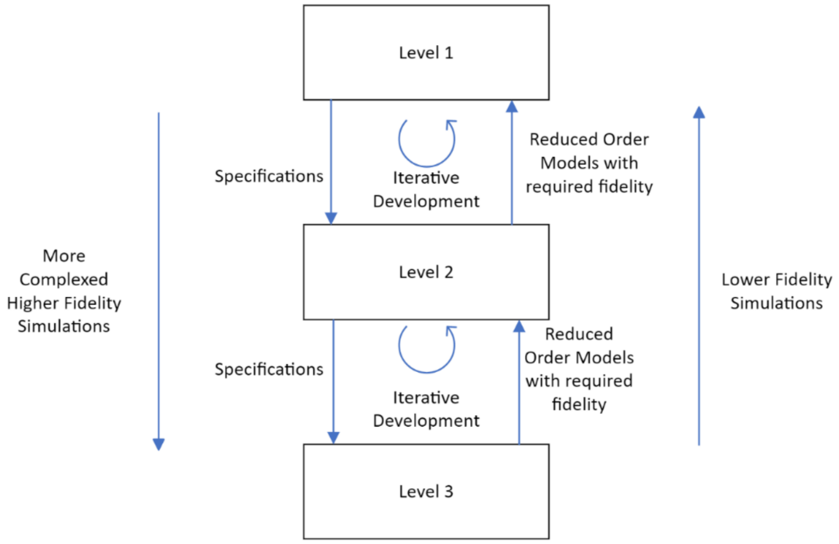


Figure 4. General Framework of Simulation Hierarchy

This basic framing of possible interactions between the different simulations is then applied to the various simulation levels found in Chapter 2 to obtain the interaction highlighted in Figure 5.

3.3 Describing Interaction Between Different Levels of USAR Robotic Simulation

Figure 5 summarizes the interactions between the various layers of simulation, with the USAR Concept of Operations (CONOPS) Simulation being the highest-level simulation and sub-component simulation being the lowest-level simulation. Drawing analogies with DoD's Model Hierarchy, the CONOPS simulation would consist of higher-level campaign simulations that would simulate the overall effectiveness of various types of USAR operations.

This USAR CONOPS Simulation drives the needs of the multi-robot system. For the USAR CONOPS Simulation to have more accurate performance details, however, it would need more accurate models on the performance capabilities of the system, and this could be provided by the Multi-Robot Mission Modeling and Analysis Simulation. Likewise, the simulation would then drive the system performance specifications that the engineering team would need to achieve from the system, and a higher fidelity simulation that includes more accurate physical models and the actual use of algorithms and/or hardware interfaces would be carried out. The lowest level will be the robot subcomponent level simulation, where the individual sub-components of the robots will be simulated with the main aim of providing a design that can be physically manufactured or actual code that would be used in the system.

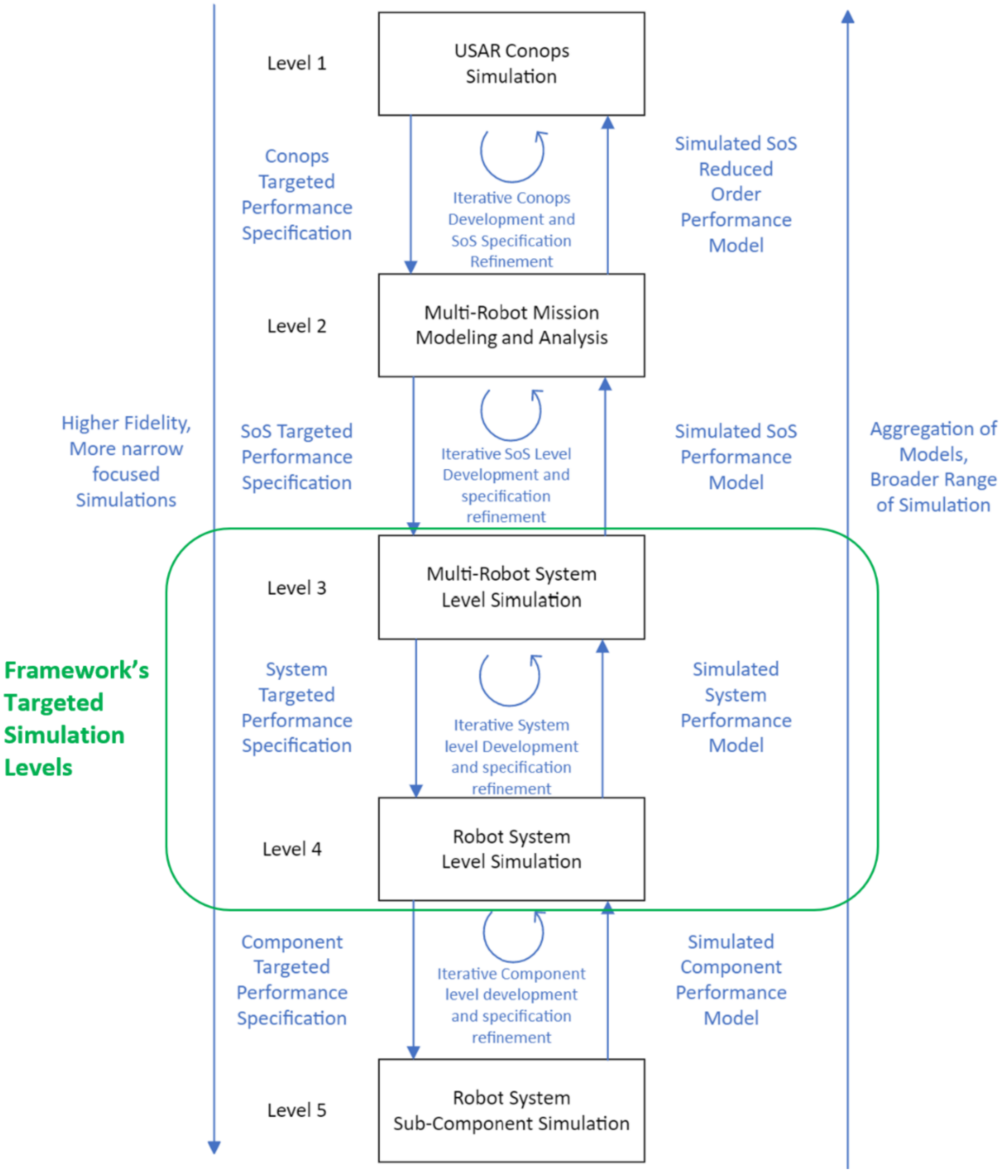


Figure 5. Interaction Between Various Levels of Simulation

3.4 Description of Motivation for Various Levels of Simulation

Based on the literature survey, Table 8 highlights examples of possible stakeholders' need as well the elaboration of the interactions between the different levels of simulations.

Table 8. Levels of Simulation, Stakeholders and Purpose

Levels	**Type of Simulation**	**Stakeholders Involved**	**Purpose for Simulation**	**Information Feedback**	**Benefits for Stakeholders**	**Example of Simulation**
1	CONOPS Simulation	USAR Users, Program Managers	To simulate the effectiveness of the chosen USAR solutions used to address a specific USAR scenario. The USAR Operation Simulation could simulate the use of several different USAR tools, with the use of a multi-robot system could be one of them.	Feedback Between Levels 1 and 2: Provide specific parameters needed by the multi-robot system, found in level 2 that would lead to a successful USAR Operation. Possible information could include coverage of search area and rate of coverage. It can obtain higher fidelity performance results of a proposed multi-robot system that was simulated in level 2. This allows better fidelity in modelling the overall USAR Operational performance. Possible information could include the expected speed of coverage of a given search area.	The authorities responsible for USAR operations can derive the overall effectiveness of USAR Operations based on the various systems chosen for the operation. The Project Managers would be able to better understand how their multi-robot system impacts the overall USAR Operational effectiveness.	Mainly military CONOPS simulations are found during literature review [56], [57]
2	Multi-Robot System Modeling and Analysis	System Engineers, Program Managers	To simulate the overall performance of a chosen configuration of a multi-robot system in terms of meeting the requirements needed by the USAR Operation.	Feedback Between Levels 2 and 3: Provide level 3 simulation with specific configurations of robots needed to be developed in order to meet the mission objectives. Possible information includes robot types and their combined performance targets.	The Project Manager will have a better understanding of how their products perform, hence allowing them to provide potential users with more accurate information. The System Engineer would be able to understand how the system can meet the mission requirements and possibly perform better-informed tradeoffs at the engineering levels.	MBSE design for UAV Swarm Modeling [29]

3	Multi-Robot System Level Simulation	System Engineers, Program Managers, Component Engineers	To identify individual robot specifications needed to meet the mission objectives, as well as the development of engineering components that can only be carried out at the multi-robot system level. Such developments could include multi-robot various coordination algorithms and mission operator HMI at a multi-robot system level.	Feedback Between Levels 3 and 4: Provide level 4 simulations with robot-specific target parameters that are needed by the multi-robot system to meet mission objectives. Possible information includes the robots' speed and endurance and sensor and actuator performance. It can also obtain higher fidelity performance parameters on how individual robot types perform based on level 4 robot system simulation. This will aid in multi-robot algorithm development as reduced-order robot models may be needed to simulate multi-robots simultaneously.	Speeds up the development process for the engineers by reducing the physical testing of physical prototypes needed for system-level multi-robot system development. The engineers would be able to test the performance of their development on a wider range of scenarios that may not be feasible in physical testing.	- HMI simulation and interaction with SAR teams [21] - Simulation of multi-robot algorithm [5]
4	System	System Engineers, Component Engineers	To identify components or robot architecture needed to meet robot performance targets, as well as to develop various system-level components such as robot control and perception algorithms.	Feedback Between Levels 4 and 5: Provide level 5 simulations with the targeted component specification. Possible information could include mechanical structure properties, actuator requirements, or electronic component requirements. It can also return higher fidelity information on the performance that the component is able to offer. An example will include the overall stiffness of the chassis that could affect the control algorithms.	The Project Manager will have a better understanding of how their products perform, hence allowing them to provide potential users with more accurate information. The System Engineer would be able to understand how the system can meet the mission requirements and possibly perform better-informed tradeoffs at the engineering levels.	- Simulation of robotic Systems to work in USAR environments [58]
5	Sub-Component	Component Engineers	To identify design requirements needed for the development or manufacturing of sub-component, as well as to virtually test the sub-component designs being created.		To speed up and potentially lower the cost of sub-component development by virtually testing the designs, hence reducing the need for physical design iterations.	- Mechanical Structural Simulation - Electronics Design Simulation - Flight Controller Design

3.5 The focus of the Thesis Simulation Framework

Table 8 highlights the wide arrange of possible simulations that can be carried out throughout the development of the Heterogenous Multi-Robotics Simulator. This thesis will not aim to describe a framework that will address the end-to-end simulation of all the five possible levels of simulation found in that table. The thesis will focus on levels 3 and 4, which cover development of system level and multi-robot system level development.

Based on simulation levels 3 and 4, the primary stakeholders of the simulation framework would be the system engineers together with the engineers who are tasked with creating algorithms, system-level design decisions and testing, as well as Human-Robot Interactions at the robotic system level and above. Simulations carried out at these two levels will be vital in providing better resolution and understanding of the entire multi-robot simulation capabilities, which would be important for the Project Manager in providing possible solutions for potential users. It will also help in providing clearer specifications to component engineers carrying out simulations at level 5 so that they would be able to design subcomponents that meet the system’s needs.

3.6 Description of Hypothetical SAR Use-Case for Framework Application

USAR was chosen as it currently presents challenges that are still being worked on today and are currently still an open area of research. Hence, it will be a good area to base the framework on, as there would be a need to incorporate existing technologies as well as to develop new systems.

The framework would be applied to a hypothetical use-case which forms a combination of the various USAR events found in Table 3. The hypothetical earthquake that could happen in a city with a mix of high-rise and low-rise buildings causes widespread damage to the city. Table 9 highlights the combination of various needs needed from the hypothetical use-case based on the past events highlighted in Table 3.

Table 9. Broad Needs of Example USAR Scenario

USAR Operation Needs	How Robots Could Address Needs
Navigate unstable building structures	Ability to enter buildings without placing rescue personnel in harm's way.
Identify dangers such as hazardous substances/gases, or unstable structures	Ability to screen the disaster area for hazards without placing rescue personnel in harm's way.
Locate and assess trapped victims in collapsed structures	Ability to access spaces that are not accessible to rescue personnel with conventional tools.
Manipulate rubble in a collapsed building environment	Ability to clear dangerous obstacles within unstable structures without placing rescue personnel in harm's way.
Fast and Efficient Response	Ability to search wider areas without facing the same fatigue faced by rescue personnel.

3.7 Summary

This chapter introduces the various levels of simulations and how they interact with each other to allow for the development of a SAR multi-robotic system. These simulations can range from a high-level simulation that deals with the USAR concept of operations all the way to the component level of a robotic system. This thesis does not attempt to cover all five simulation levels. Instead, it will focus on simulation levels 3 and 4, and the broad needs highlighted in Table 9 will drive the framework described in the following chapter. The introduction of the USAR operation that will be used to demonstrate the framework will also drive the needs in chapter 5.

4 Introduction to Framework

4.1 Introduction

This chapter aims to introduce a generic flow that one can take to formulate a basic multi-robot simulation architecture based on high-level USAR concepts and needs. The block diagrams generated within this chapter are based on the author's interpretation of the literature survey highlighted in Chapter 2, followed by analysis tools that will be used to analyze the simulation architecture.

The first step is to identify the various possible interactions between the main components in an Urban Search and Rescue (USAR) operation. The operational needs will then be used to formulate the multi-robot system as well as the individual robots in the robotic system. The needs formulated from the multi-robot, as well as the individual robotic systems, will then be used to formulate the simulation architecture. Once the diagrams have been generated, the DSM can then be built and analyzed. The connections in the diagram represent the flow of information or needs from one entity to the next.

Figure 6 summarizes the process of deriving the Robot Simulation System Architecture that would address the given USAR operation needs. The USAR operation architecture will drive the needs of the multi-robot system, and the needs (Tables 10, 12 and 14) will propagate down towards formulating the final robotic simulation architecture. Tables 11, 13, and 15 highlight how these architectures contribute to addressing the USAR operation needs.

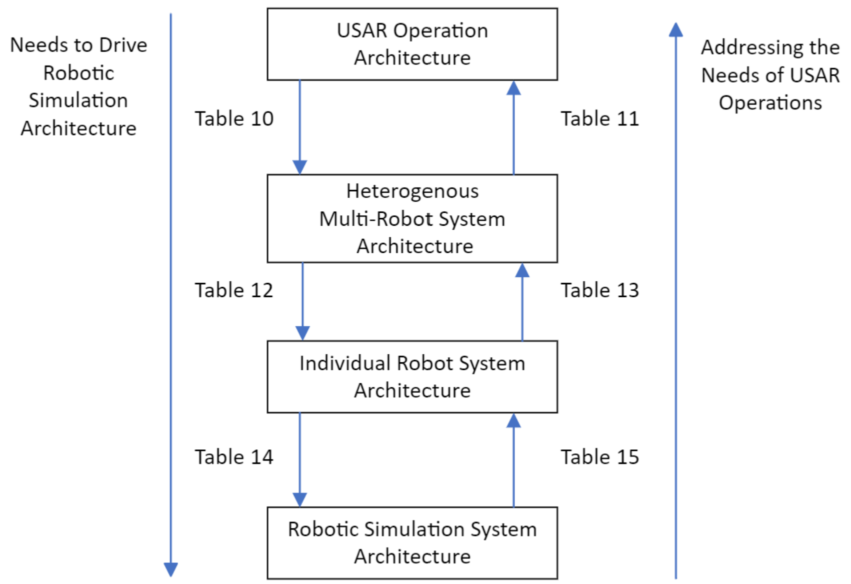


Figure 6. Deriving Robotic Simulation Architecture of USAR Operation Needs

4.2 Urban Search and Rescue Interaction

Figure 7 represents the high-level USAR architecture that shows how the different USAR components interact with each other based on the literature survey carried out in Chapter 2.

The SAR management team would be central in coordinating the various efforts in the search and rescue operation by coordinating with the local authorities and providing feedback on the ongoing situation, as well as coordinating the efforts within the various search and rescue teams. The logistical team could place constraints on the types of robots that would be used, such as the size and the types of robots that could be used at the disaster sites. For clarity, the Search and Rescue teams are split according to their functions. The personnel carrying out the search would need to assess the disaster area and locate potential victims, while the personnel carrying out the rescue would need to be able to assess and extract the victims. The medical teams would need to carry out triage of the victims, as well as render first aid.

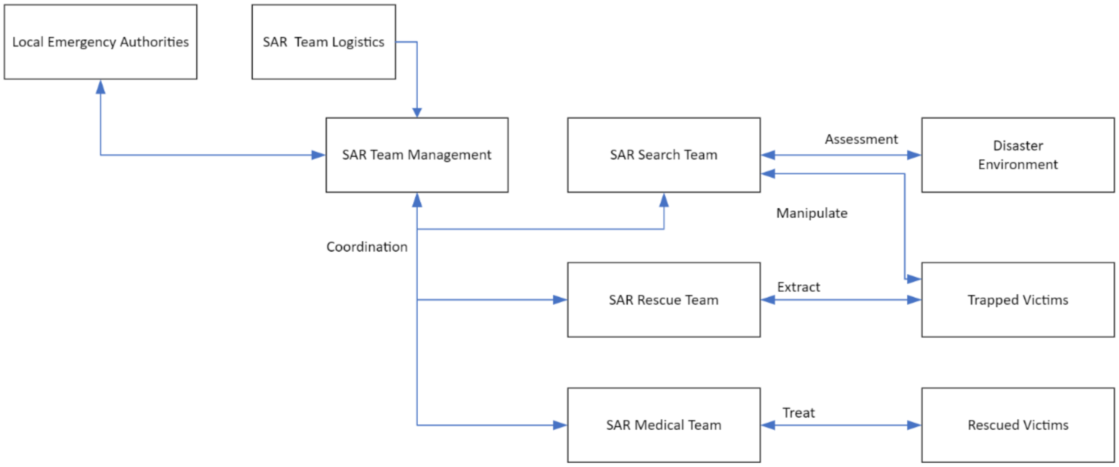


Figure 7. Block Diagram USAR Component Interaction

Based on this basic interaction, there could be a wide variety of possible needs that could be derived. However, due to brevity, a subset of these possible needs would be considered for this example. Table 10 highlights the broad needs that would be used to drive the architecture of multi-robotic systems.

Table 10. USAR Operation Needs

USAR Operation Needs
Need for coordination between the various teams and the management
Need for situational awareness from higher authorities.
Need for assessment of the disaster area
Need to manipulate trapped victims
Need to manipulate disaster site.
Need to treat and aid rescue victims
Maintain the Safety of all personnel involved in the effort.

4.3 Multi-Robot System Interaction

Figure 8 represents the high-level block diagram of the proposed multi-robot system architecture that was conceptualized to meet the USAR operation needs. The mission Ground Control Station (GCS) will be central in the coordination between the robots and the human operator. The operator interacts with the robots by controlling their behavior through the inputs given to the coordination algorithm, while the authorities would be able to monitor and receive feedback through the operator or possibly direct feedback from the mission GCS, as shown in the DARPA Sub-T Challenge [5]. The robots can be controlled in a centralized or decentralized manner to achieve the task required by the operator. Each robot in the system could be controlled by a human operator that deployed the robot and would need to interact with various elements in the USAR disaster site. Communication between the robots and the GCS would be critical in enabling coordination between various systems.

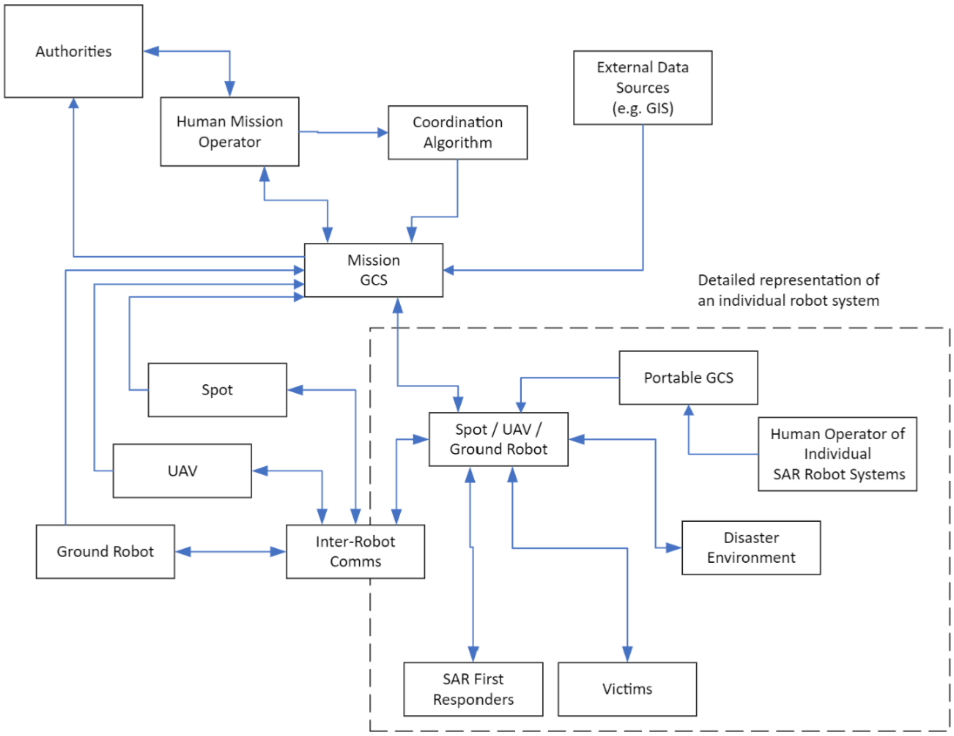


Figure 8. Block Diagram of Multi-Robot System Interactions

Table 11 highlights how the multi-robot system architecture addresses the USAR operation needs.

Table 11. Multi-Robot System Addressing USAR Operation Needs

USAR Operation Needs	Multi-Robot System Architecture meeting the Needs
Need for coordination between the various teams and the management	The authorities that oversee the operation can communicate with the operator or possibly have information feeds directly from the mission GCS.
Need for situational awareness from higher authorities.	The mission feed from the GCS can be a one-way communication that allows the authorities to monitor the USAR operation and interact with the operators if needed.
Need for assessment of the disaster area	The various interactions with the robot and the external entities are represented by connections and will be detailed in the robot architecture.
Need to manipulate trapped victims	
Need to manipulate disaster site.	
Need to treat and aid rescue victims	
Maintain the Safety of all personnel involved in the effort.	

Table 12 summarizes the needs for the multi-robot system based on the architecture presented above. These high-level needs would need to be addressed by the robot architectures so that each robot would be able to function effectively as part of the multi-robot system. The main needs would be communication, interaction with human controllers, and interaction with disaster sites.

Table 12. Multi-Robot System Needs

Multi-Robot System Needs
Communications between robots and GCS
Multi-robot coordination behaviors
Human-Robot System Interaction between Robot system and Mission Control Operators, Individual Robot Operators, SAR Responders, Victims
Ability to interact with disaster sites and victims.

4.4 Robotic Systems Interaction

Figure 9 represents the high-level block diagram of the proposed individual robot system architecture that was conceptualized to meet the multi-robot system needs. The algorithms are housed inside the computing solution of the robot. These computing solutions are often split up into higher-level computing solutions that house higher-level intelligence, such as perception and mapping, and interfacing with higher-level sensors to feed information to these algorithms. The lower-level computer will contain algorithms that are responsible for the locomotion of the robot with the help of various sensors that senses the robot’s state and its immediate environment. Lower-level computers will also often connect to various actuators that will act as propulsion for the robot. Another key component will be communications, which is responsible for communication between other robots and the ground controllers that controls the overall behavior of the robot. The external interactions with the robots will include its disaster environment, which consists of the victims and the physical environment, as well as the personnel on the ground, which includes other rescue personnel and its operator.

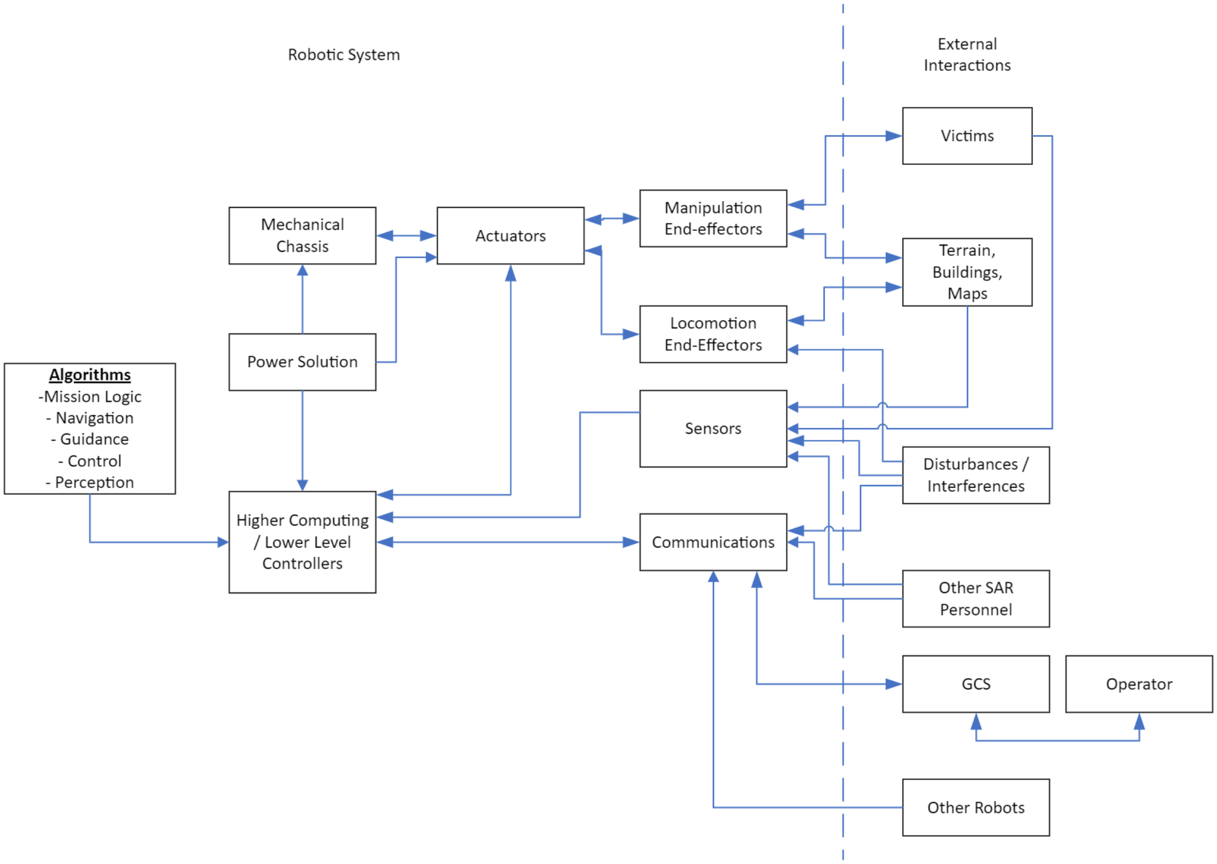


Figure 9. Diagram of Robot System Needs Interactions

Table 13 highlights how the robot system architecture addresses the needs presented by the multi-robot system.

Table 13. Individual Robotic System Addressing Multi-Robot System Needs

Multi-Robot System Needs	How this architecture meets the needs
Communications between robots and GCS	Communications will be handled by the computing solution, hence allowing the algorithms running on the computing solution to interface with other entities on the communication networks.
Multi-robot coordination behaviors	These will be handled by the mission logic algorithms housed in the higher-level computers.
Human-Robot System Interaction between Robot system and Mission Control Operators, Individual Robot Operators, SAR Responders, Victims	The humans who are directly within the disaster site will be sensed by the robot sensors, while the humans handling the robots will do so through robot communications.
Ability to interact with disaster sites and their victims.	The robot has sensors to sense the disaster site, as well as manipulators to interact.

Table 14 describes the general needs based on the robotic system described in the diagram above. They can be in terms of simulated models, or they can be represented by the physical objects during simulations, such as having actual hardware components for HWIL or having Operators in the loop.

Table 14. Robotic System Needs

Robotic System Needs
Limitations imposed by Computing Solution
Physical Interaction of mechanical elements with the environment
Interactions of sensors with their environment.
Communications between robotic systems and external sources.
Need to model physical entities found in USAR disaster sites such as victims and physical disaster areas.
Consideration of human collaborators or controllers such as other SAR personnel or robot operator [19], [41], [59]
Representation of other robotic systems
Development of algorithms

4.5 Robotics System Simulation

The diagram in Figure 10 shows how the robotic simulation system would cater to the various types of simulation, which are mainly the Hardware-in-the-Loop and Software-in-the-Loop testing and simulation. Hence, the components seen in the diagram below could be a mix of actual hardware and software components. The interactions represent the direct connection of information from one component to the other. Hence a modification of one component could potentially affect the other. The simulation interface component forms a critical component that bridges the simulation environment with the actual hardware and software to be developed, while the various components that make up the simulation engines may be used to output information to simulate the information received by different robot sensors. For example, the rendering engine would be used to generate images used by camera sensors, while the physics engine will be used to generate collisions between objects that can be sensed by the simulated robotic force sensors.

This diagram also highlights the importance of considering other supporting functions that are needed from the simulation other than that of the robot and how these functions could possibly interact with the robot components. For example, there would be a need to create models of the robot and possibly collect and store data related to the robot.

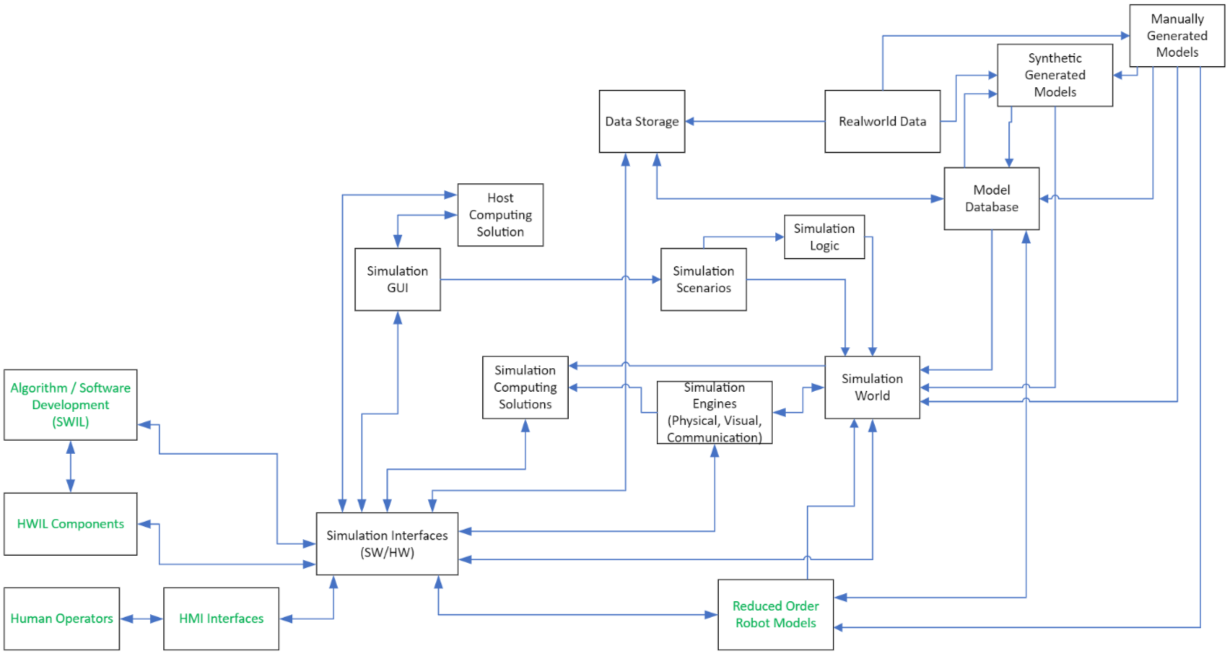


Figure 10. Diagram of Robot Simulation Interactions

Table 15 concludes the tracking of general needs from the USAR concept to the final simulation architecture. The diagram generated represents a high-level representation of a single robotic system. The representation of several different robotic systems using this diagram could result in a large, complicated diagram that could be difficult to read. Hence, the diagram should be split up accordingly to highlight the simulation of each key system that would be required in the Robotic SoS Simulation capability. This will be shown in the application of the framework highlighted in the subsequent chapter.

Table 15. Simulation Architecture Addressing Robotic System Needs

Robotic System Needs	**How this architecture meets the needs**
Limitations imposed by Computing Solution	HWIL components can be simulated to take into account the performance of actual computing hardware.
Physical Interaction of mechanical elements with the environment	Software that performs a simulation of real-life physical phenomena such as physical interactions, visual feedback or communication effects will be represented by the simulation engines.
Interactions of sensors with their environment.	
Communications between robotic systems and external sources.	
Need to model physical entities found in USAR disaster sites such as victims and physical disaster areas.	Various virtual models will be combined and represented in the simulation world. This virtual environment will present a sandbox in which the robot will interact with its environment.
Consideration of human collaborators or controllers such as other SAR personnel or robot operator	The human operators can interact with the simulation through an HMI interface
Representation of other robotic systems	Reduced order robots will be used to represent other virtual robots that will make up the multi-robot system that is carrying out the USAR operation.

4.6 Analysis of Simulation Architecture Using Engineering Tools

The analysis of the robot simulation architecture can be done using a DSM. This tool serves two purposes. (1) It can represent complex simulation architectures that would be easier to understand the interaction between various components as compared to large directional graph networks. Different modularizations can be explored to improve the organization of the components. This is especially useful if the simulation architecture contains many components and interactions. (2) For graphs in which the interactions can be represented with numerical values, calculations can be carried out on the DSM to uncover insights about the architecture. For this case, the System Readiness Levels will be calculated to gain insights into the readiness of the simulation capability described by the proposed simulation architecture. The three main steps below highlight how simulation architecture can be analyzed using DSM and SRL calculations.

4.6.1 Building the Robot Simulation Architectures

The multi-robot simulation setup can be targeted to simulate two or more robots with varying simulation needs and interactions. It will be beneficial to build the robot simulation architectures separately as the interactions between the various robot components and the simulation components could become large and complex if they are represented in a single graph. The various robot simulation architectures can then be combined in a DSM. This method will allow for the scaling up of the number of robots that can be represented in this simulation framework.

4.6.2 Building the DSM

The various robot simulation architectures can then be represented in a single DSM. Care needs to be taken to make sure that there is no duplication of common elements that appear throughout the various robot simulation architectures. An example of such common elements would be the presence of the physics, rendering and communication simulation engines.

Once the DSM has been populated with the components with its directional connections, the DSM can be clustered using various DSM analysis tools or through manual clustering. The aim of clustering is to be able to group components with similar interactions or properties. The clustering will allow for better organization between engineers working on building up the simulation capabilities while also allowing for a more intuitive analysis of the readiness levels of broad sets of component groups.

4.6.3 Populating the TRL and IRL Levels Within DSM

The TRL and IRL values are populated based on the description found in **Error! Reference source not found.** and

Table 7, which contains the description of how these readiness levels can be estimated. For a newly conceptualized simulation system, such as the system conceptualized in Chapter 5, the IRL and TRL are based on the available information that can be readily found in the proposed sets of components being used.

4.6.4 Calculating SRL

Once the TRLs and IRLs of the DSM have been populated, the SRL can then be calculated. Depending on the desired insights to be gained on the system readiness of the components, the SRL can be analyzed at the component level or be aggregated by averaging over clusters, groups, or the entire robot simulation architecture. Higher levels of aggregation allow decision-makers to gain convenient insights into the overall readiness levels of the evaluated simulation architecture at the expense of having a resolution on the contributing factors. This thesis chooses to analyze the SRL at the component level so that higher-resolution insights can be observed based on the different technical decisions made.

4.7 Conclusion

This chapter outlines the systematic buildup of the multi-robot simulation architecture starting from its targeted operational needs. The final architecture can then be analyzed using DSM and SRL tools that would enable decision-makers to appreciate the readiness levels of a proposed simulation architecture or to monitor the capability build-up in an existing simulation architecture. The framework can now be used to formulate a multi-robot simulation architecture based on the hypothetical USAR scenario described in Chapter 3.

5 Application of Framework

5.1 Introduction

The problem definition provided at the end of Chapter 3 lays the foundation for the choice of robotic systems to be used in this framework. A mixture of three types of robotic systems will be used for this exercise to highlight the unique qualities of having different strategies for acquiring robots.

5.2 General Robot Types Chosen to Evaluate for Application of Framework

Three different classes of robots will be evaluated for the purpose of the framework. The robots are chosen with the main aim of highlighting how these selections would impact the application of the framework as well as the system readiness levels evaluated by the DSM. The robot types are also chosen to broadly address the USAR needs highlighted in Chapter 2. Examples of such robots are as follows:

Quadruped Robot: These robots have four legs and can traverse across uneven terrain while maintaining their balance. These qualities would allow them to be deployed in various areas in which conventional wheeled or tracked vehicles would otherwise have difficulties, such as the ability to traverse in manmade environments with obstacles such as stairs to the ability to traverse uneven or slippery surfaces [60]. These robots could be used to traverse dangerous and unable terrains during a USAR, such as potentially unstable buildings or rubbles to areas which could potentially be toxic to humans. Some notable examples of these robots would include those from Boston Dynamics Spot, Unitree Robotics and Ghost Robotics [60]. These robots can be customized and purchased from the manufacturers as a complete solution, or a basic model could be purchased with the ability to integrate customized payloads [61].

Unmanned Aerial Vehicle: These aerial robots would be able to survey large areas of terrain in a shorter amount of time as compared to their ground-based counterparts, hence providing authorities with better situational awareness of the disaster sites [29]. This can be achieved by mapping the areas with various different types of sensors, such as LiDAR, as well as using having the ability to use its cameras to identify objects over a wider field of view as compared to a ground robot [31]. Similar to the quadruped robots, these robots can also be customized or purchased from manufacturers as a complete solution or have the ability to integrate customized payload options on the basic drone platform. Some notable examples include popular mainstream drone manufacturers such as DJI, which sells proprietary drone hardware that can be interfaced with to integrate custom payloads [5], to vendors such as UAV Systems, which integrate open-source hardware to provide drones with basic flight capabilities.

Snake Robots: These robots make use of multiple articulated joints to traverse their environment as opposed to traditional tracks, wheels, or legs [62]. The main goal of these robots is to traverse environments that have space constraints. There are several snake robots that are being developed by researchers in various universities, such as the snake robot developed and used by CMU in several disasters. Compared to the two robotic systems above, snake robots are not widely commercially available. Hence, it would be assumed that extensive in-house development would be needed by any organization that wishes to mature and deploy such technologies as compared to being able to acquire the robotic platform through COTS products.

Table 16 tracks how the choice of these robots addresses the broad needs highlighted in Table 9.

Table 16. Meeting Needs of Example USAR

Needs	**Quadruped Robot**	**Unmanned Aerial Vehicle**	**UGV Robots**
Navigate unstable building structures	x		X
Identify dangers such as hazardous substances/gases, or unstable structures	X		X
Locate and assess trapped victims in collapsed structures			X
Manipulate rubble in a collapsed building environment	X		
Fast and Efficient Response		x	

5.3 Different Strategies for Acquiring Robotic Systems

There are three main ways that the robotic system could be acquired. Each of these strategies will be used to evaluate the following robotic systems in the following sections of this chapter. The following points briefly highlight the rationale of each strategy.

(1) Purchase the entire solution from a vendor. This strategy can be used to acquire platforms from vendors when the vendors are able to provide or customize platforms to meet operational needs or when it is too costly to build the platform from the ground up. This strategy could reduce the overall cost of acquiring the platform and shift engineering resources to develop other areas of the multi-robot system that cannot be purchased. The drawback of this strategy would be the dependency on the robot vendors to provide models or software interfaces that work with the chosen simulation setup.

(2) Purchase a robotic platform from a vendor and integrate it on customized payloads. This strategy involves the purchase of the base robot platform from vendors and subsequently adding on a customized

payload that would be developed in-house. This strategy can be used when the vendors do not provide the exact robotic solution needed but provide a reliable and cost-effective robot platform for payload integration. The drawback of this strategy would be the dependency on the robot vendors to provide simulation and interface models of the base robot platform.

(3) Develop the robot from the ground up using COTS components. This strategy involves building the entire robot from the ground up using existing components that are readily available to purchase. This strategy allows for the construction of a robot to meet specific needs that may not be fulfilled by existing commercially available robots. Building a robot from the ground up requires the largest amount of engineering resources for the development of the robot hardware, as well as the modelling of the robot within the simulation environment.

5.4 Selected Quadruped Robot

The proposed system to be selected for the quadruped robot will be the SPOT robot from Boston Dynamics. This robot system is assumed to be acquired entirely from the vendor without any engineering effort needed to customize the robot. Instead, the vendor will provide the customer with a choice of several different robot configurations. The robot shown in Figure 11 represents the SPOT robot configuration used in this example. This robot configuration is one of the predefined robot configurations offered to customers [61, p. 2].

Figure 11. SPOT Robot Configuration Available from Boston Dynamic (image source: [61])

5.4.1 High-Level Hardware Architecture

The high-level architecture of the SPOT Robot system is shown in Figure 12 below. The main engineering efforts that are expected after acquiring the robot platform are as follows:

- Multi-Robot coordination algorithms
- Grasping and interacting algorithms using the robot arm
- Development of specialized perception algorithm using the camera payload.
- Autonomous interactions with human entities

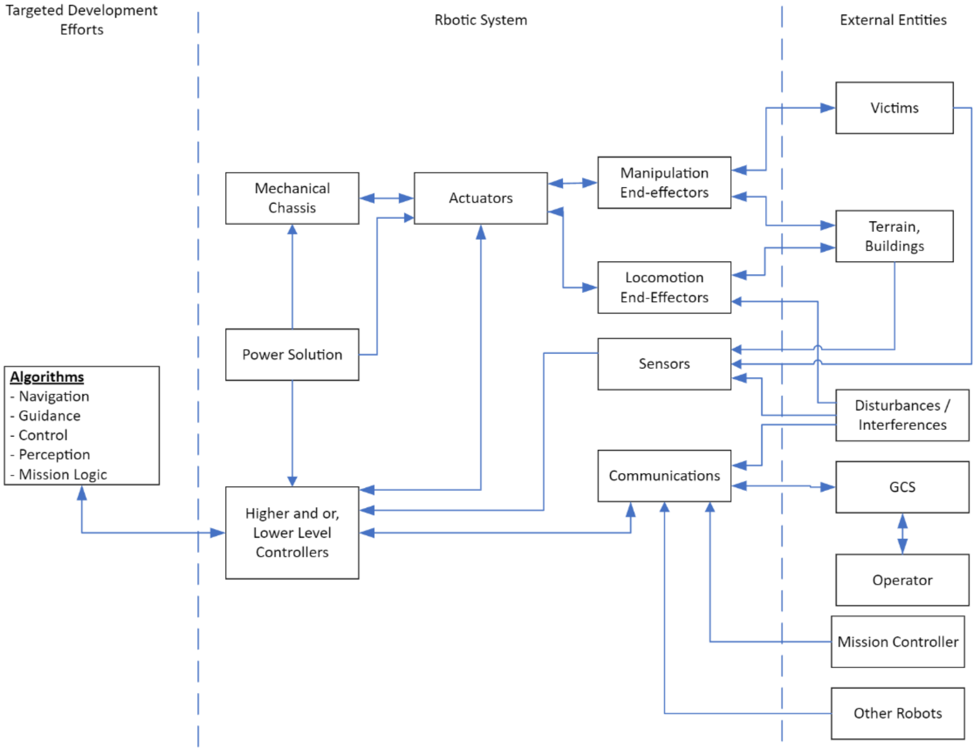


Figure 12. High-level Hardware Architecture of Quadruped Robotic Platform

5.4.2 Proposed Simulation Architecture

Figure 13 outlines the proposed quadruped architecture. The main engineering efforts would be the software that resides on the robot, as well as the human interaction interfaces with the robot. As such, most of the robots could be simulated except for human-machine interactions. However, the creation of robot models for use in the simulation could require information from the vendors. This dependency could make it difficult to develop simulation models if they are not readily available from the vendors.

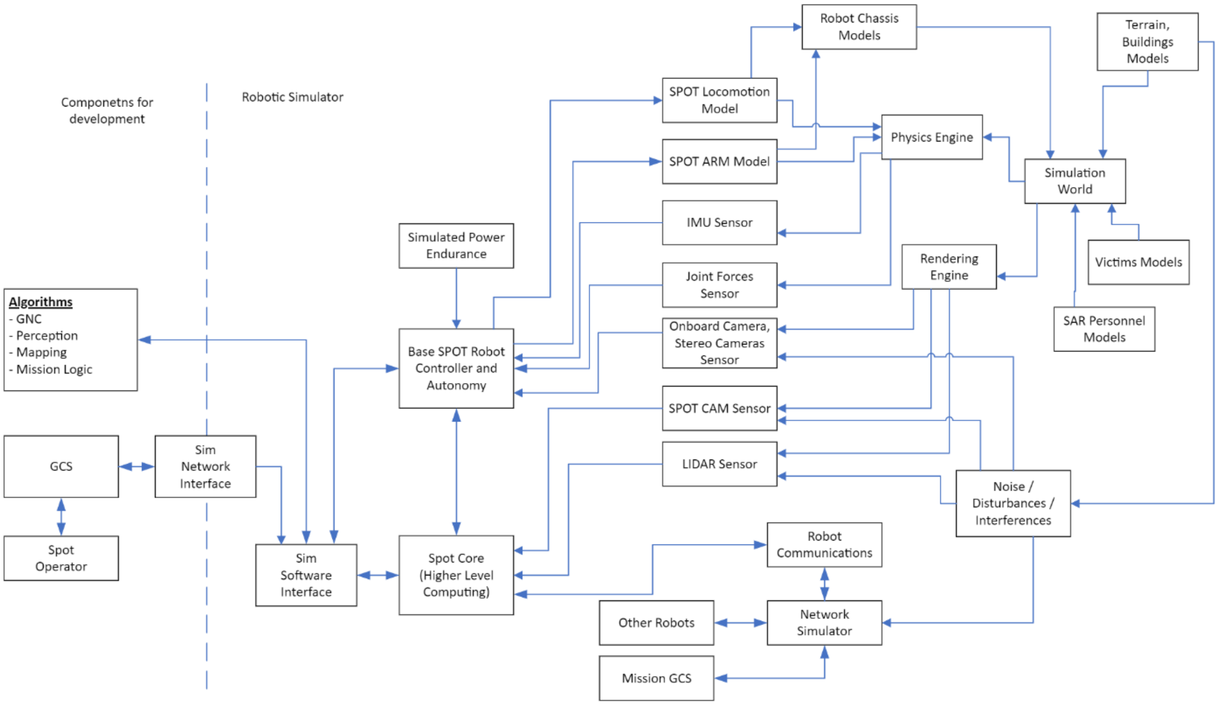


Figure 13. High-level block diagram representing main components of interest to simulate Quadruped Platform

5.5 Selected Unmanned Aerial Vehicle

The proposed UAV platform that is chosen for this example would be the DJI M100 or M600 drones that were available from DJI such as the drone seen in Figure 14. This strategy allows for the acquisition of reliable robotic platforms that would be able to integrate customized payloads that the manufacturers may not offer. Hence, the engineering teams developing the UAV robots would be able to focus on the payload development and overall higher-level intelligence of the UAV platform.

Figure 14. Example of DJI M600 integrated with customized payloads (image source: [63])

5.5.1 High-Level Hardware Architecture

The high-level architecture of the UAV Robot system is shown in Figure 15. The main engineering efforts that are expected after acquiring the robot platform are as follows:

- Multi-Robot coordination algorithms
- Payloads such as the integration of sensors or the higher-level computing platform.
- Perception, mapping tasks and other sensing tasks based on those sensors.
- Interaction with a human operator

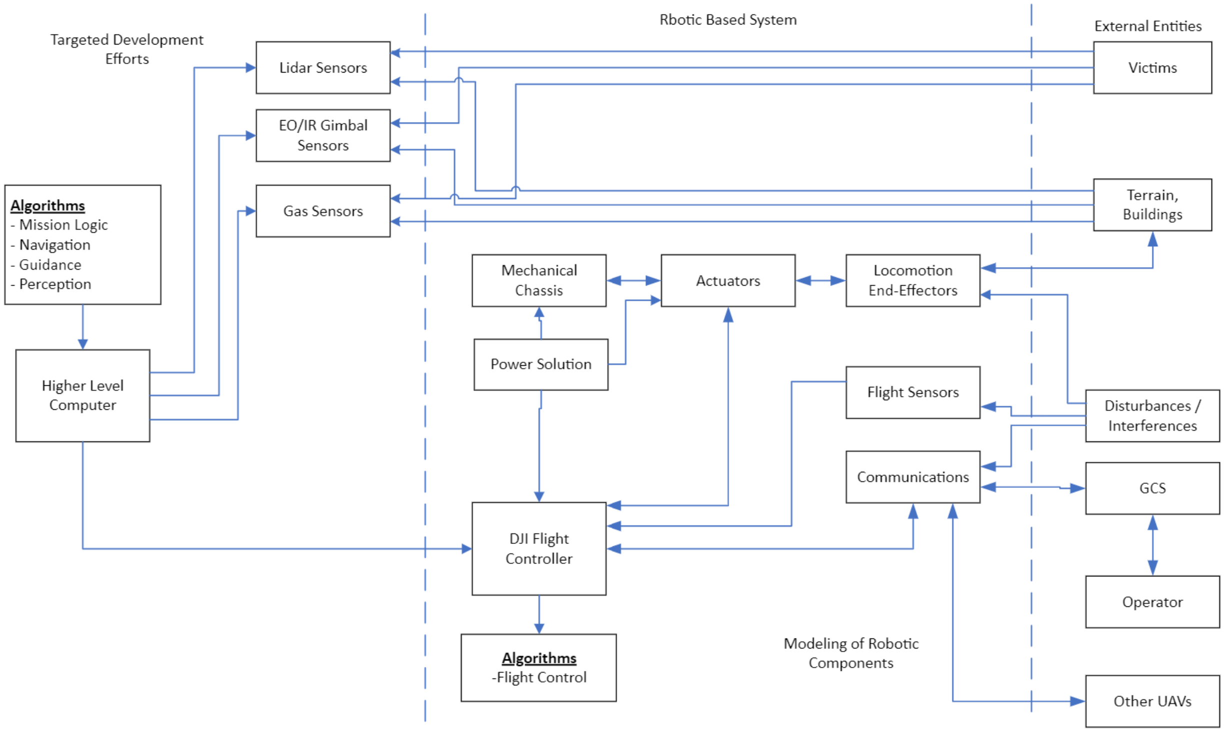


Figure 15. High-Level Hardware Architecture of UAV Platform

5.5.2 Proposed Simulation Architecture

The proposed UAV simulation architecture is shown in Figure 16. The base robot can be simulated as there would not be a need to develop basic UAV capabilities such as flight controls. However, there would be a need to obtain such models from the vendor for simulation. Payloads that need to be developed for the UAV could interact with the simulation setup through HWIL simulation, hence allowing the developed hardware to be used directly on the robot during physical testing. This will bridge the gap between simulation and the testing of the physical systems.

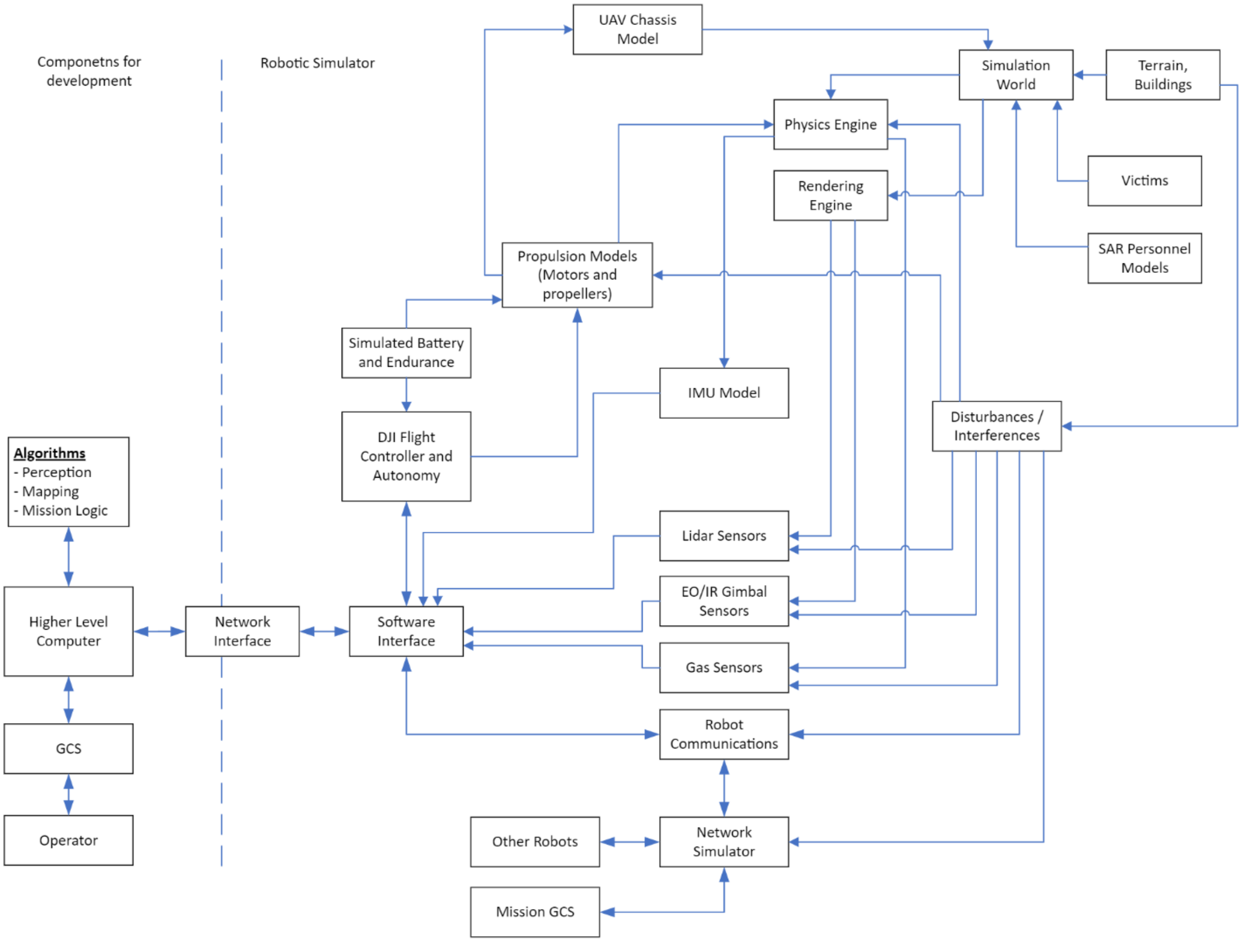


Figure 16. High-level block diagram representing main components of interest to simulate UAV Platform

5.6 Selected Unmanned Ground Vehicle

The type of robot to be developed for the unmanned ground vehicle would be a snake robot that would be able to navigate within confined spaces found in a disaster site. Some examples would include the modular snake robot developed by the Biorobotics Lab at Carnegie Mellon University or the Guardian S robot platform that is commercially available from Sacros as shown in Figure 17. For this example, it is assumed that no commercial robot would be able to satisfy the projected operational needs, and a robot will need to be built from the ground up using COTS components.

Figure 17. Examples of robots with snake-like maneuverability (left image: [64] right image: [65])

5.6.1 High-Level Hardware Architecture

The high-level architecture of the Ground Robot system is shown in Figure 18. The main engineering efforts that are expected after acquiring the robot platform are as follows:

- Multi-Robot coordination algorithms
- Physical hardware design that would achieve its goal of entering confined spaces
- Basic lower-level robot functionalities such as robot control and locomotion
- Higher-level robot functionalities such as perception to sense for survivors or to sense for hazards.
- Human-robot interaction with both the victims and the SAR personnel could be temporarily taking manual control of the robot.

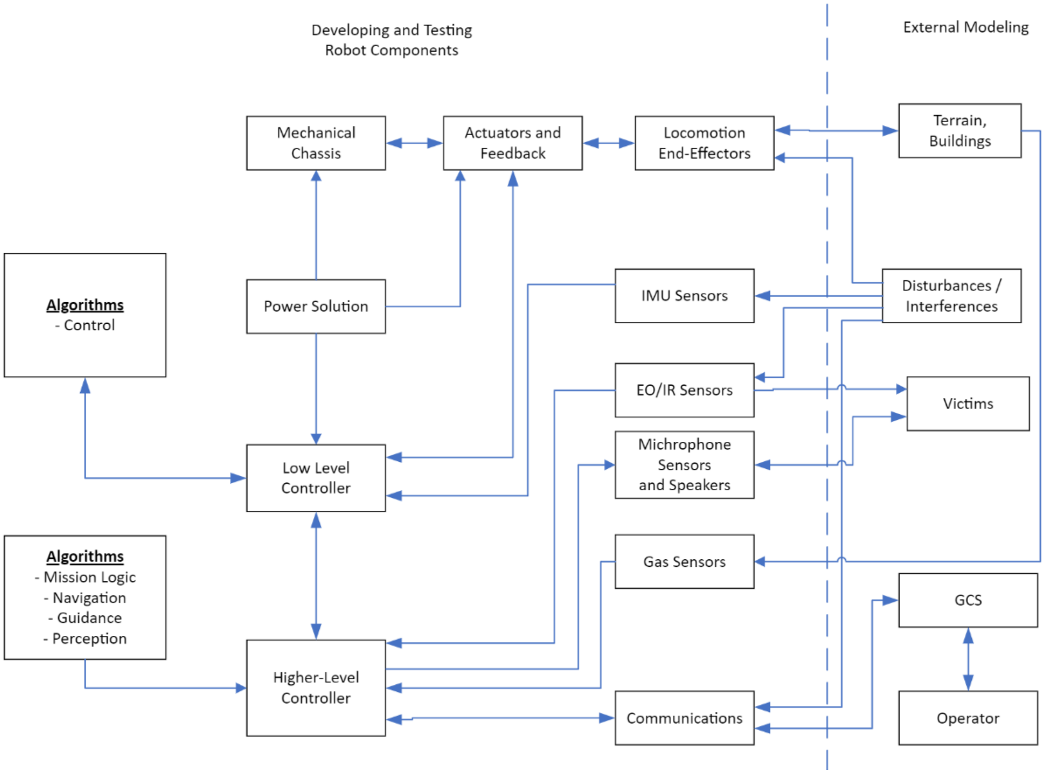


Figure 18. High-level Hardware Architecture of Ground Robotic Platform

5.6.2 Proposed Simulation Architecture

HWIL simulation could be used together with the simulation setup to reduce the need for physical testing of the robot while making sure that the chosen hardware for the robot can meet the various operational needs. In proposed architecture show in Figure 19, it is assumed that both the higher-level and lower-level computers will undergo HWIL testing and development. However, the sensors and actuators of the robot will need to be simulated. The interface into the simulation engine can also be tailored to fit the interface needs of the controllers. For example, the lower-level computer could be fed signals that emulate actual sensor interfaces, hence cutting down the development effort needed to use the controller on the physical robot.

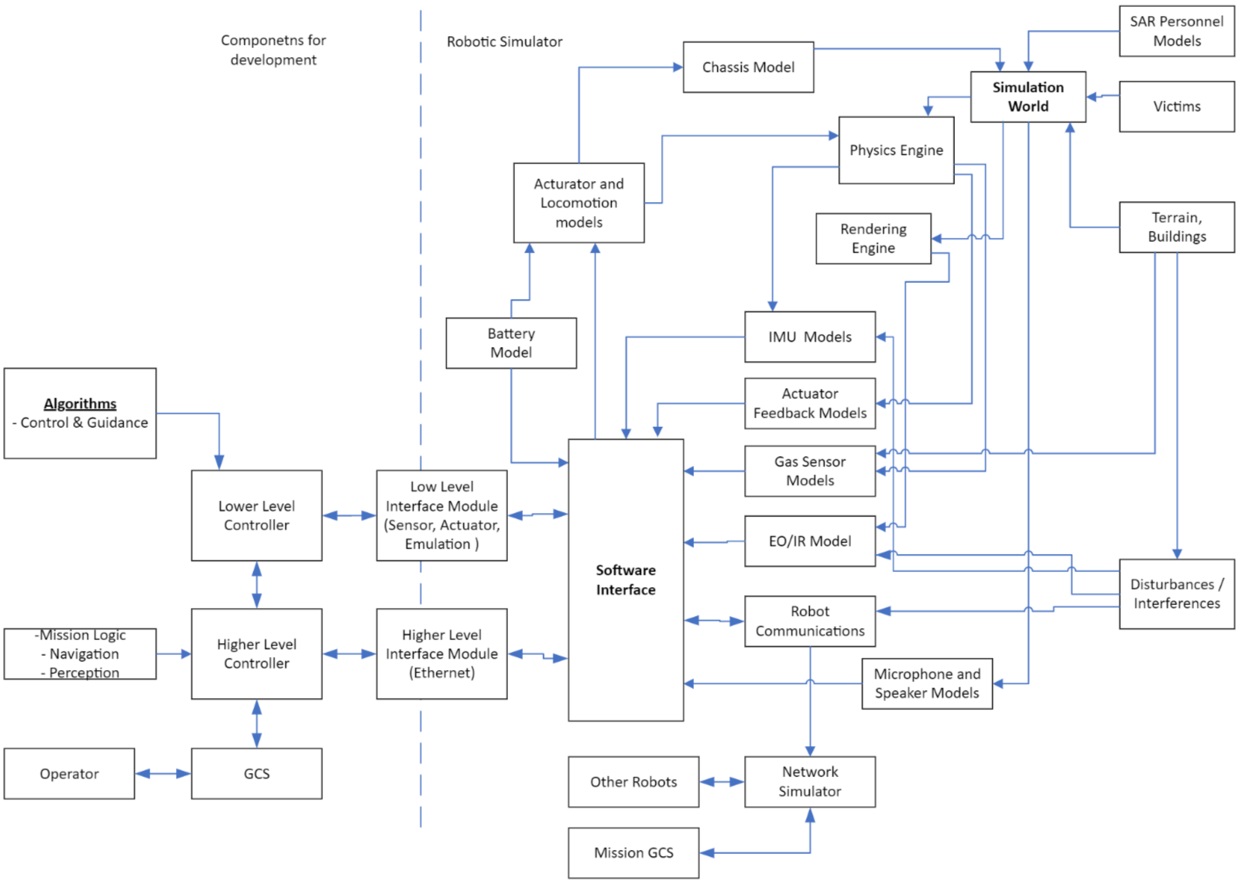


Figure 19. High-level block diagram representing main components of interest to simulate Ground Platform

5.7 Selected Multi-Robot Configuration

The multi-robot system is a fully customized system that would be developed to meet the projected needs of the range of USAR operations that the system is designed to address.

The use of a central ground control station (GCS) is chosen for this multi-robot system example, together with the ability of the robots to communicate with each other to allow for the coordination of their actions. This will allow for the type of algorithm that requires a centralized node to allocate tasks while allowing the robots to interact with each other to perform the allocated tasks [30]. The human operator in control of the multi-robot system would be able to interact with the robots through communications from the central ground control station. Higher-level authorities in charge of coordinating the USAR operation would be able to receive updates from the GCS and give inputs to the human operator if operational needs change.

5.7.1 High-Level Hardware Architecture

The main engineering efforts that are expected after acquiring the multi-robot system, as shown in Figure 20, are as follows:

- Centralized multi-robot coordination algorithms are housed within the mission GCS, as well as the decentralized multi-robot coordination algorithms that occur between the robots.
- Creation of the mission GCS and its interaction with human operators.

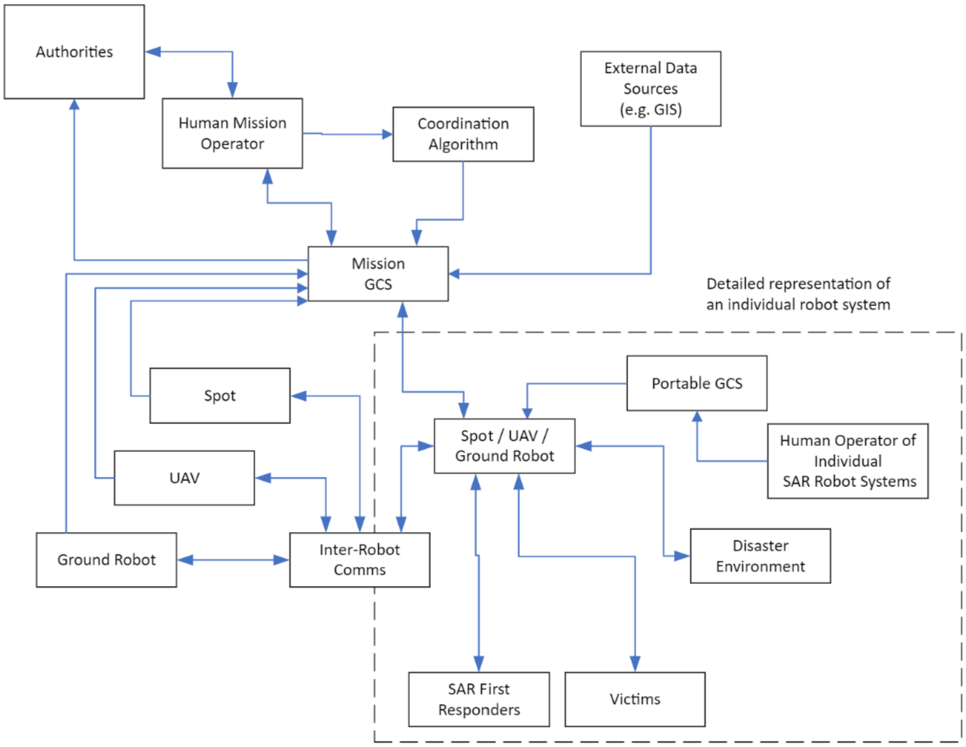


Figure 20. High-level Hardware Architecture of Multi-Robot System

5.7.2 Proposed Simulation Architecture

Figure 21 shows the simulation of multiple robots needed to develop multi-robot coordination algorithms could require the use of reduced order robots as high-fidelity robots could exceed the computing solution's simulation capabilities. For example, a high number of robots requiring high-fidelity physical interaction with their environment would slow down the simulation, as described by [66]. However, the simulation of large numbers of UAVs without the need to interact with their physical environment or the need to render camera sensor images could be done as it requires lesser computing resources. This simulation architecture also highlights how a detailed single robot interfaces with the multi-robot simulation.

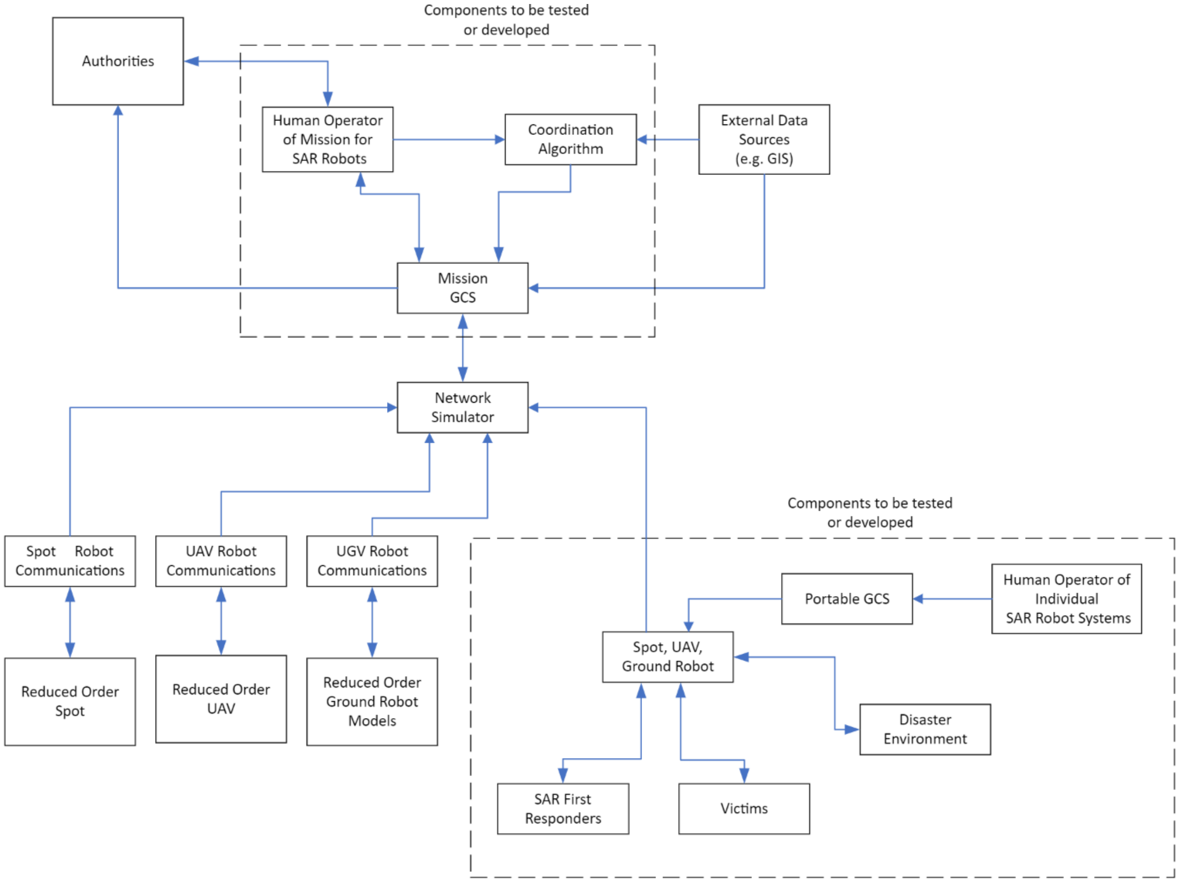


Figure 21. High-level block diagram representing main components of interest to simulate Multi-Robot System

5.8 Choice of Simulation Engines and Software Middleware

The simulation engine Gazebo, together with the middleware ROS, was chosen for this evaluation as they are extensively used in the academic development of algorithms and robotic systems. As noted in Chapter 2, since they are widely used, both have good support for a wide array of robotic systems. In this case, the SPOT robot models and there are existing models for the DJI drone. Hence these factors will bring up the readiness levels of the interfaces and the simulation components, hence impacting the IRL and the TRL levels covered in the section below.

5.9 Building the DSM for Analysis:

5.9.1 Building the DSM.

Figures 13,16, and 19 highlight the three different robotic systems that were chosen to be part of the multi-robot system. Figure 21 highlights the component required to link these individual robotic systems together. Within these four figures, the common simulation elements can be extracted to represent the simulation system that will be used to simulate the multi-robot system. Examples of such common elements include the simulation engines (Rendering, Physics and Communication) and the simulation world with its objects. Specialized simulation components found in Figure 10, such as Manually Generated Models, Synthetic Generated Models, and Data Storage, were also included in the simulation system. Based on the figures described above, with the addition of the common simulation elements, five groups of components can be derived as follows: (1) Simulation System, (2) Quadruped Platform Simulation, (3) UAV Platform Simulation, (4) UGV Platform Simulation, (5) Multi-Robot Simulation.

The elements represented by these five groups are represented as rows and columns in the DSM. The connection represents the direct flow of information from one component to the other. For example, there are bi-directional flows of information from the algorithm component that is being developed and the software interface that connects to the simulated robotic components, while the models that feed into the simulation world can be considered as a one-way informational flow as the simulation world makes use of this model information to create the world and no information is being fed back into the models. The DSM is then built according to the method that has been outlined in Chapter 2, and further analysis of the DSM was carried out, which includes the modularization of components and the analysis of the system readiness level. The DSM with the elements of the multi-robot simulation system is shown below, with a one-directional flow of information. The CAM2 tool was then used to explore the potential clustering of the components.

5.9.2 Clustering the DSM.

There are two possible strategies of modularization; (1) to run an algorithm to modularize all the components of the DSM and to tweak the outcome of this modularization, (2) to maintain the integrity of these five systems and to modularize the components within these systems to form subcomponents.

<u>Method 1:</u> Modularization was carried out on all 83 elements of the DSM. This method could be desirable as it will allow for the clustering of components that could be more optimal as compared to the original clusters that are based on the five main system groups. However, an effort will need to be made to make sense of intuitive reasoning on why those components in the clusters go well with each other and how they interact. Subsequently, the organization could also reorganize to make use of the new clustering.

<u>Method 2:</u> The components will be grouped according to the five main groups. The clustering is then done within the five main groups. This will allow the integrity of the five main systems to be kept while optimizing the organization of components within the systems. Hence, this could be easier to manage at

an organizational level as there could be teams working on each system, while at the sub-team level, the teams would be able to organize their efforts more efficiently due to the new clusters. The outcome of the clustering can be seen below.

Method 2 was chosen as this would be more intuitive for interpretation. The overview of the interactions between the submodules can be clearly seen. In the overview of the DSM shown by Figure 22, we can see that the interaction of the interfaces mainly happens between the main simulation system and the simulated robotic system. By design, there are no interactions between the various simulation modules within the simulated robotic components. More implications of these interactions are discussed in the analysis section below. Once the modularization has been cone, the TRL and IRL values can then be populated.

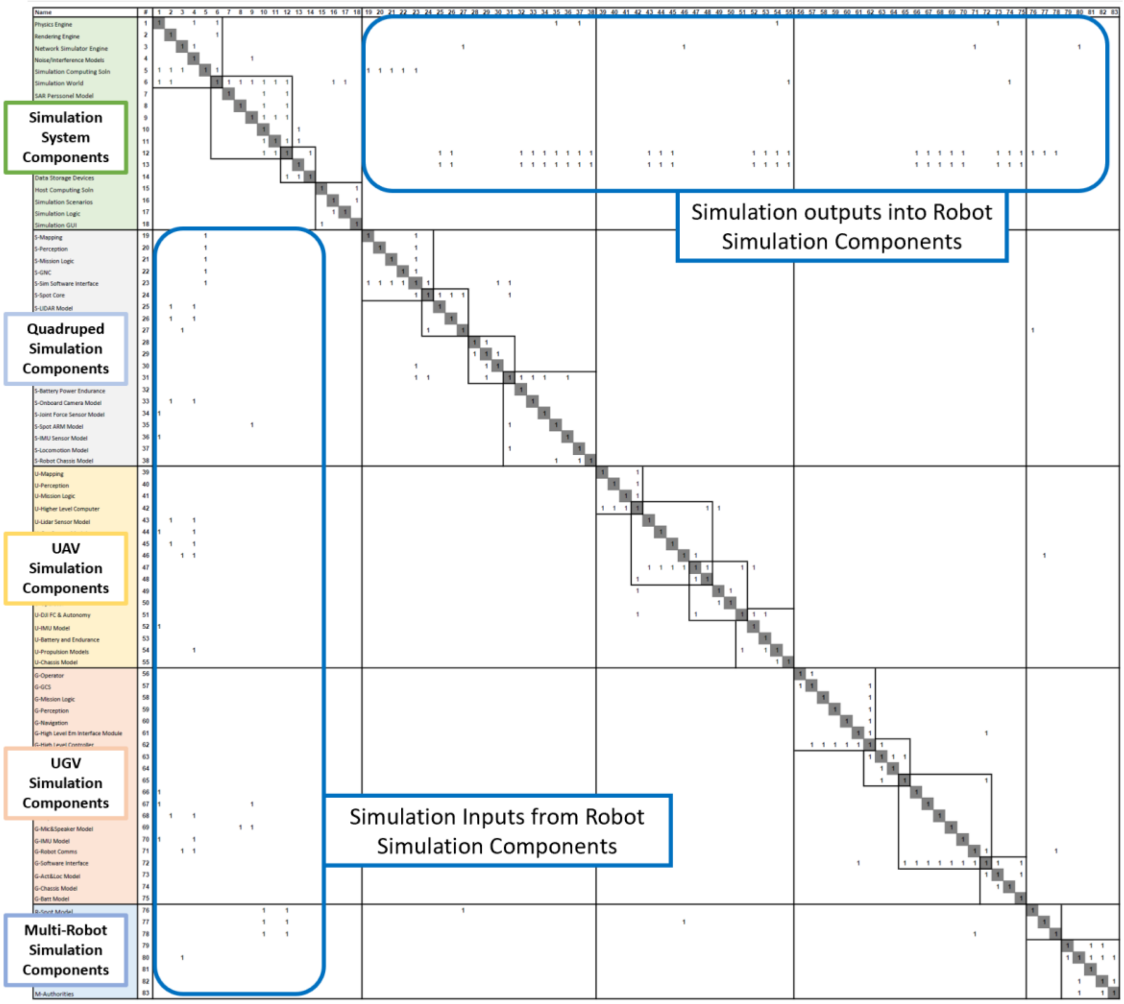
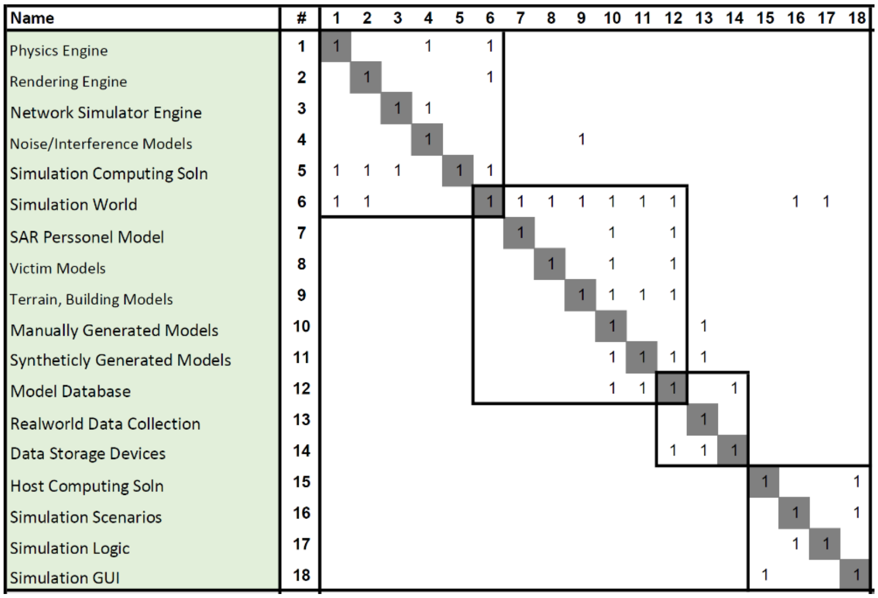


Figure 22. Overall DSM After Clustering

5.9.2.1 *Clusters Formed from Modularization*

Simulation Clustering. The simulation system can be clustered into four groups. Group 1 would deal mostly with the simulation engine and the required computing resources. Group 2 deals with the models to be created and used in the simulation engine. Groups 1 and 2 share the overlap of the simulation world as the models that make up the simulation world interact strongly with the physics and the rendering engine. Group 3 makes up the data storage components, while group 4 contains the components that are needed to set up the simulation environment.

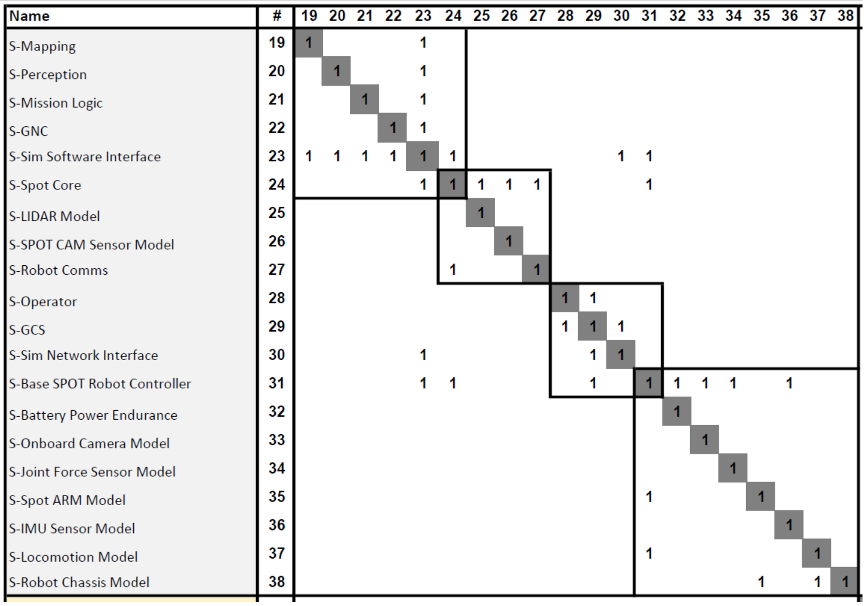


Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Physics Engine	1	1			1		1												
Rendering Engine	2		1				1												
Network Simulator Engine	3			1	1														
Noise/Interference Models	4				1					1									
Simulation Computing Soln	5	1	1	1		1	1												
Simulation World	6	1	1				1	1	1	1	1	1	1				1	1	
SAR Perssonel Model	7							1			1		1						
Victim Models	8								1		1		1						
Terrain, Building Models	9									1	1	1	1						
Manually Generated Models	10										1			1					
Syntheticly Generated Models	11										1	1	1	1					
Model Database	12										1	1	1		1				
Realworld Data Collection	13													1					
Data Storage Devices	14												1	1	1				
Host Computing Soln	15															1			1
Simulation Scenarios	16																1		1
Simulation Logic	17																1	1	
Simulation GUI	18															1			1

Figure 23. Clustering of Simulation Engine Components

The robotics system, in general, can be clustered into four main groups. However, each robot system has slightly different clusters.

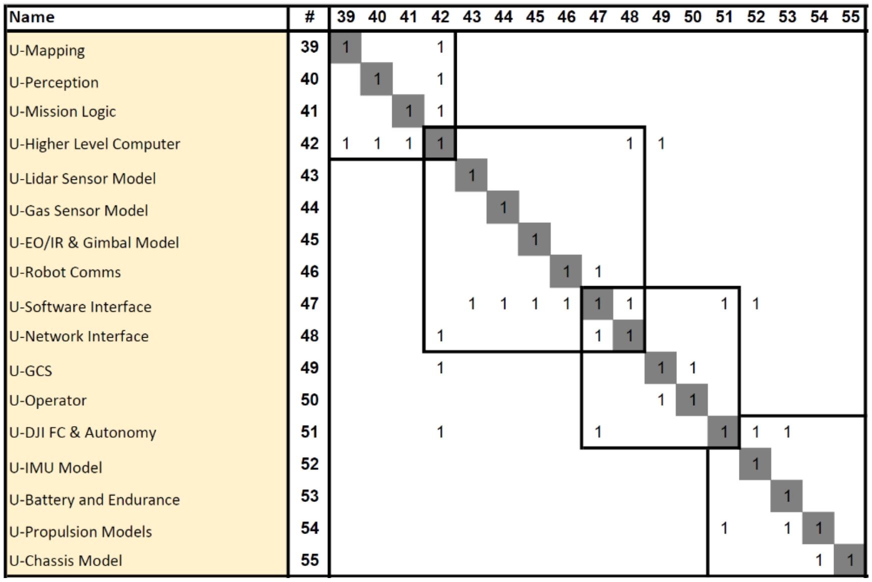
SPOT Robot Clustering. For the SPOT robot system, Group 1 consists of the algorithms that are used within the robotic system as well as the Spot Core, which is the higher-level computing hardware. This component overlaps with Group 2, which mainly consists of the simulated sensors that will interact with the higher-level computing hardware. Group 3 would consist of the GCS and its operator, which shares an interface with the lower-level SPOT controller as well as the simulation components used to simulate the base SPOT robot.



Name	#	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
S-Mapping	19	1				1															
S-Perception	20		1			1															
S-Mission Logic	21			1		1															
S-GNC	22				1	1															
S-Sim Software Interface	23	1	1	1	1	1	1						1	1							
S-Spot Core	24					1	1	1	1	1				1							
S-LIDAR Model	25							1													
S-SPOT CAM Sensor Model	26								1												
S-Robot Comms	27						1			1											
S-Operator	28										1	1									
S-GCS	29										1	1	1								
S-Sim Network Interface	30					1						1	1								
S-Base SPOT Robot Controller	31					1	1					1		1	1	1	1		1		
S-Battery Power Endurance	32														1						
S-Onboard Camera Model	33															1					
S-Joint Force Sensor Model	34																1				
S-Spot ARM Model	35													1				1			
S-IMU Sensor Model	36																		1		
S-Locomotion Model	37													1						1	
S-Robot Chassis Model	38																	1		1	1

Figure 24. Clustering of SPOT Robot Components

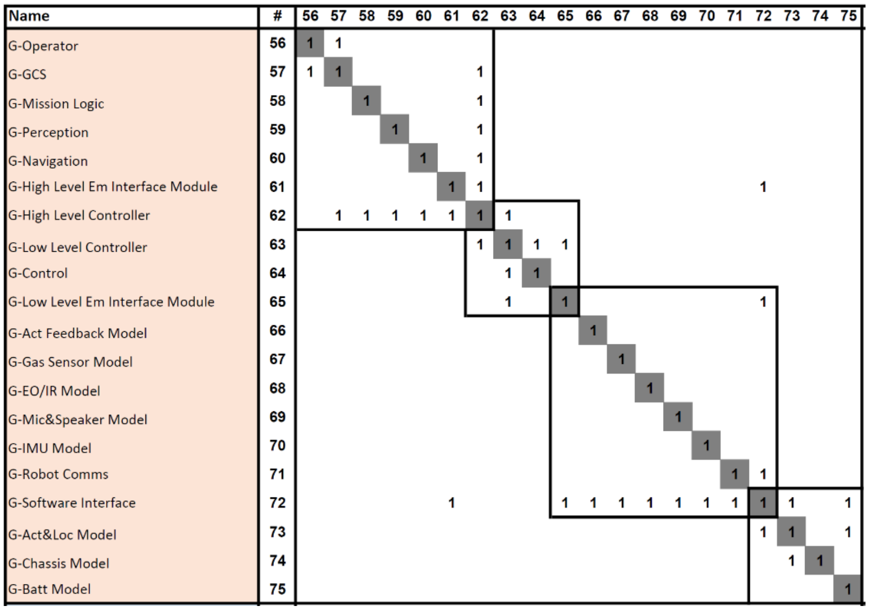
<u>UAV Robot Clustering.</u> Group 1 of the UAV system consists of higher-level mission algorithms and the higher-level computing solution. This overlaps with group 2, which mainly consists of the sensor models used by the higher-level algorithms and computing solutions. Group 2 and group 3 overlap at the network and software interfaces and mainly consist of the GCS and operator. Group 3 overlaps with group 4 at FC and Autonomy module. Group 4 mainly consist of lower-level flight control systems for the UAV.



Name	#	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
U-Mapping	39	1			1													
U-Perception	40		1		1													
U-Mission Logic	41			1	1													
U-Higher Level Computer	42	1	1	1	1						1	1						
U-Lidar Sensor Model	43					1												
U-Gas Sensor Model	44						1											
U-EO/IR & Gimbal Model	45							1										
U-Robot Comms	46								1	1								
U-Software Interface	47					1	1	1	1	1	1			1	1			
U-Network Interface	48				1					1	1							
U-GCS	49				1							1	1					
U-Operator	50											1	1					
U-DJI FC & Autonomy	51				1					1				1	1	1		
U-IMU Model	52														1			
U-Battery and Endurance	53															1		
U-Propulsion Models	54													1		1	1	
U-Chassis Model	55																1	1

Figure 25. Clustering of UAV Robot Components

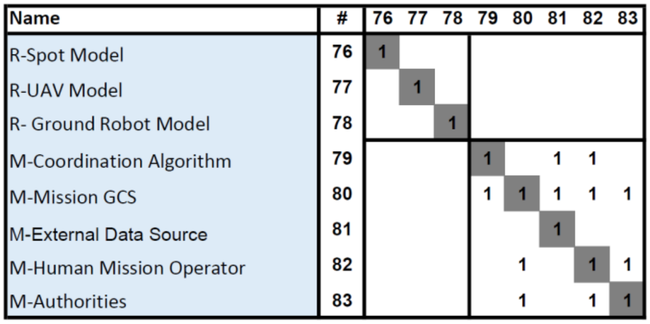
Ground Robot Clustering. For the Ground System, Group 1 contains the higher-level computing solution together with the higher-level algorithm. However, unlike the other two robot systems above, it will be beneficial to group the GCS and the operator into the first group. Group 2 mainly of the robot controllers and the interface modules, while Group 3 consist of the sensor models used by the higher-level controllers, and Group 4 consist of the lower-level control models and interfaces.



Name	#	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
G-Operator	56	1	1																		
G-GCS	57	1	1					1													
G-Mission Logic	58			1				1													
G-Perception	59				1			1													
G-Navigation	60					1		1													
G-High Level Em Interface Module	61						1	1										1			
G-High Level Controller	62		1	1	1	1	1	1	1												
G-Low Level Controller	63							1	1	1	1										
G-Control	64								1	1											
G-Low Level Em Interface Module	65								1		1							1			
G-Act Feedback Model	66											1									
G-Gas Sensor Model	67												1								
G-EO/IR Model	68													1							
G-Mic&Speaker Model	69														1						
G-IMU Model	70															1					
G-Robot Comms	71																1	1			
G-Software Interface	72						1				1	1	1	1	1	1	1	1	1		1
G-Act&Loc Model	73																	1	1		1
G-Chassis Model	74																		1	1	
G-Batt Model	75																				1

Figure 26. Clustering of UGV Robot Components

The Multi-Robot System can be clustered into two groups; the first group will include the reduced order model robots, and the second group will include the heterogenous multi-robot coordination algorithms together with the human and computational elements that support it. Figure 27 shows that there is no interaction that happens within the first group. Instead, the interactions occur between the reduced order models and their respective robot communication components, as seen in Figure 28.



Name	#	76	77	78	79	80	81	82	83
R-Spot Model	76	1							
R-UAV Model	77		1						
R- Ground Robot Model	78			1					
M-Coordination Algorithm	79				1		1	1	
M-Mission GCS	80				1	1	1	1	1
M-External Data Source	81						1		
M-Human Mission Operator	82					1		1	1
M-Authorities	83					1		1	1

Figure 27. Clustering of Multi-Robot Simulation Components

5.9.2.2 Insights Gained from Modularization

The following points below highlights the insights gained from analyzing the modularization of the DSM.

Decoupling of Systems. Overall, it can be seen that the simulation of the three robotic systems is decoupled from each other. This would be ideal as the development of the robotic system simulation models will be able to be carried out in parallel as they have no dependencies. Hence, the DSM allows one to analyze the architecture to minimize dependencies with the aim of allowing parallel development. Hence, the main coupling will be between the robotic systems and the simulation system.

Grouping for Better Communication. The DSM represents the direct flow of information from one component to the other. Hence, the clustering of the DSM represents groups of components that have stronger connections in the flow of information. Engineers handling the interfaces and the creation of the components could be grouped together, allowing better communication when developing the interfaces. For example, we can notice that the engineers handling the sensor models as well as the robot models would need to work with or consider Data storage needs that are found within the simulation system. Another example would be the broad interaction between the sensor and mechanical models and the physics and rendering engine. This means that the outputs of the engine need to be closely coordinated at an interface-level with these sensor models.

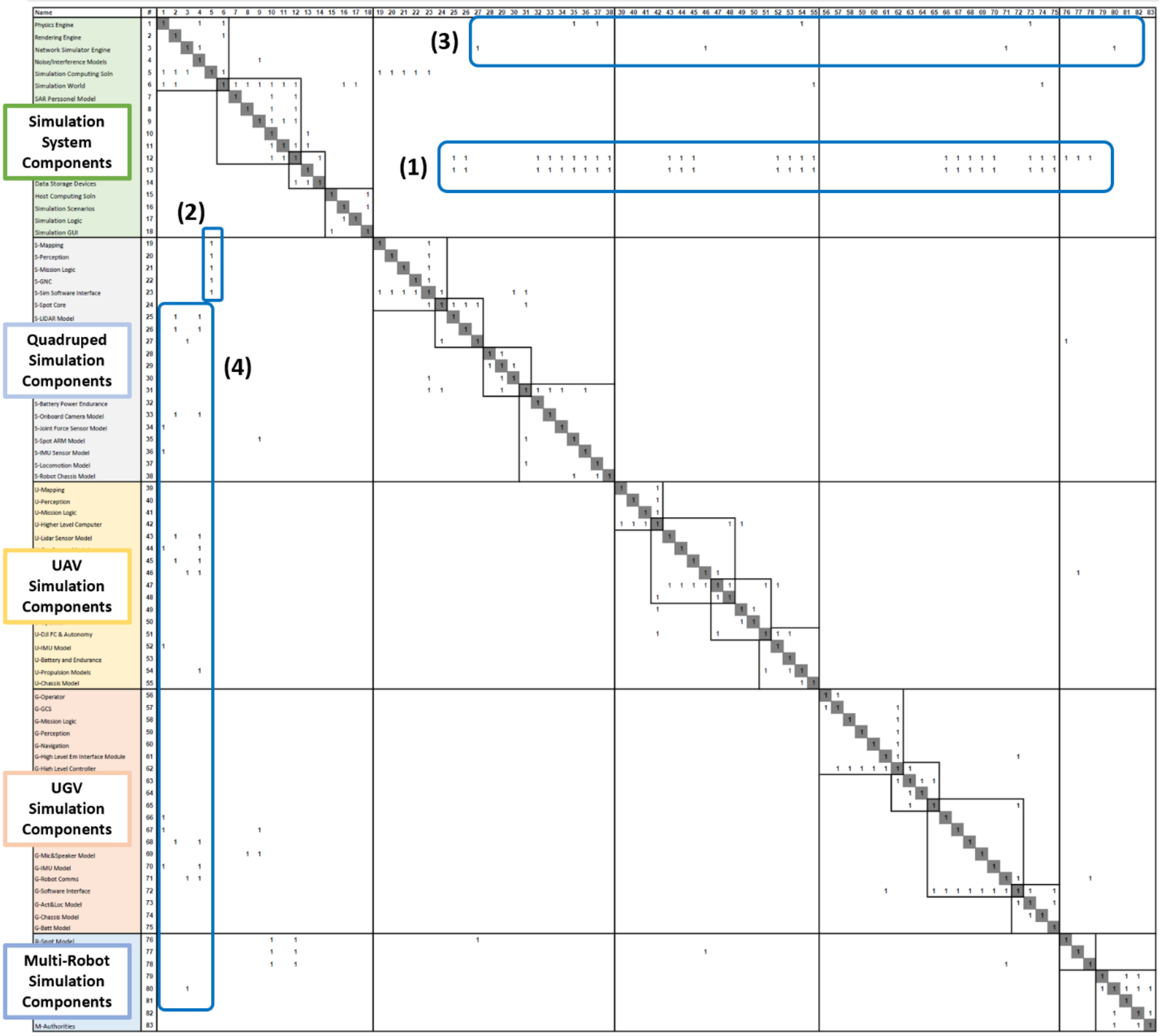


Figure 28. Interactions Between Various Groups of Components Outside of Clusters

The DSM was chosen to have five broad clusters. Hence there are various interactions that happen outside these clusters. However, the broad interactions between the various sub-groups of the clusters can be easily identified. With reference to Figure 28 and the interactions highlighted in the boxes, the points below highlight various insights gained from analyzing these interactions.

<u>(1) Data Storage and Real-World Data.</u> General interfaces of the models to the data storage and real-world data collection. These may be split up into groups depending on their modularization. As noted, it will be easy to monitor the readiness from the DSM as they are tightly clustered and not scattered throughout the DSM.

(2) Algorithms Running on Simulation Computer. It can be noted that the SPOT models are unique compared to the other simulation setups as it runs the algorithms on the simulation computer. This could mean the impact of the simulation computer on the algorithm performance needs to be considered.

(3) Robot Models Providing Inputs to Simulator Engines. As seen from the horizontal interactions, other general interactions between the simulations and the engines would be the physics engines, the simulation world and the network simulator, forming the inputs to the simulator.

(4) Sensor Models Requiring Outputs from Simulator Engines. As seen from the vertical interactions, the main interactions include the outputs of the simulation engines to the sensor models. It can also be seen that the terrain and building interact more with the models that require mechanical feedback to the sensors. This is seen more for the SPOT robot model, as well as the UGV models, as they traverse the ground and interact more with these models.

5.9.3 Population of the TRL and IRL:

The TRL and IRL metrics have been estimated based on open-source literature that is readily available to the public. The sections that follow briefly describe the assumptions made in relation to the general choice of TRL and IRL chosen to populate the DSM. In practice, these values can be populated with the inputs of subject matter experts, hence increasing the accuracy of the estimation. For clarity, the assumptions are explained based on the five main groups that were highlighted in the previous subsection, and the assignment of TRL and IRL based on these assumptions can be found in Appendix A.

5.9.3.1 Assumptions Made for Simulation Environment

Simulation Cluster: The Gazebo simulation solution has the physics engine, and rendering engine integrated [4] and is widely used for robotics simulation. Together with the software middleware ROS, the solution has a high maturity level for those components as well as their interfaces. However, Gazebo does not come with a network simulator that simulates inter-robot network communications. The network simulator engine needs to be integrated into the Gazebo simulation in a similar fashion to how it was integrated into other robot simulators [36]. Hence, the readiness of the network simulator, together with its interfaces, is of a lower level. In general, sensor noise models for chosen sensors needs to be modeled in Gazebo [67], leading to a generally low readiness level.

Models Cluster: The use of the Gazebo simulation engine for the DARPA Sub-T Challenge and other robotic challenges led to the creation of various models. Hence, being able to make use of those models would lead to higher readiness levels. However, there are various models that remain to be lacking, such as the

SAR Personnel Models, as well as any simulation logic with regard to a dynamic USAR environment. These components will need to be created according to the type of simulation scenario that was desired. There are various examples of manually generating models for use in Gazebo. However, there is lesser information on how to use real-world collected data to generate models. Lastly, there is currently no apparent workflow on how to synthetically generate various different models automatically, like how it can be done with simulators from Amazon and Nvidia.

Data Collection and Storage Cluster: There is currently a lack of information on how real work data can be incorporated and virtualized for use in the simulator. Such examples could include the use of LiDAR scans to build virtual models.

Simulation Logic Cluster: Gazebo and ROS show weaker capabilities in handling complex simulation logic and scenarios as compared to other simulation engines that allow complex scripting of scenarios using.

5.9.3.2 Assumptions Made for Quadruped Robotic Simulation

Simulation models of the SPOT robot from Boston Dynamics have been created by the company ClearPath Robotics for use in the DARPA robotics challenge [68]. Hence, most of the models and interfaces are present in the Gazebo simulator. However, the robots used in the competition were autonomous, and there was no direct control of the robots from operators. Hence, the readiness of a GCS being controlled by an operator would be low.

5.9.3.3 Assumptions Made for UAV Robotic Simulation

For the DJI flight controller that was chosen, there have been efforts in the research community to carry out both SITL and HWIL simulations. As mentioned above, the SITL route was chosen to reduce the complexity of the simulation setup. Hence the readiness of the models used to simulate the base UAV together with its flight controls would be high. However, it will not be at the same readiness level as the SPOT robotic simulations as the UAV setup will be unique to the system that needs to be built, and there will be no verification of the actual performance in real-life testing. The other components that are low in readiness levels would be the algorithms, as well as the hardware in the loop processor, as those are the components that need to be developed and customized according to the particular mission needs. The GCS and the operator will also be of low readiness level as these are highly customized components.

5.9.3.4 Assumptions Made for UGV Robotic Simulation

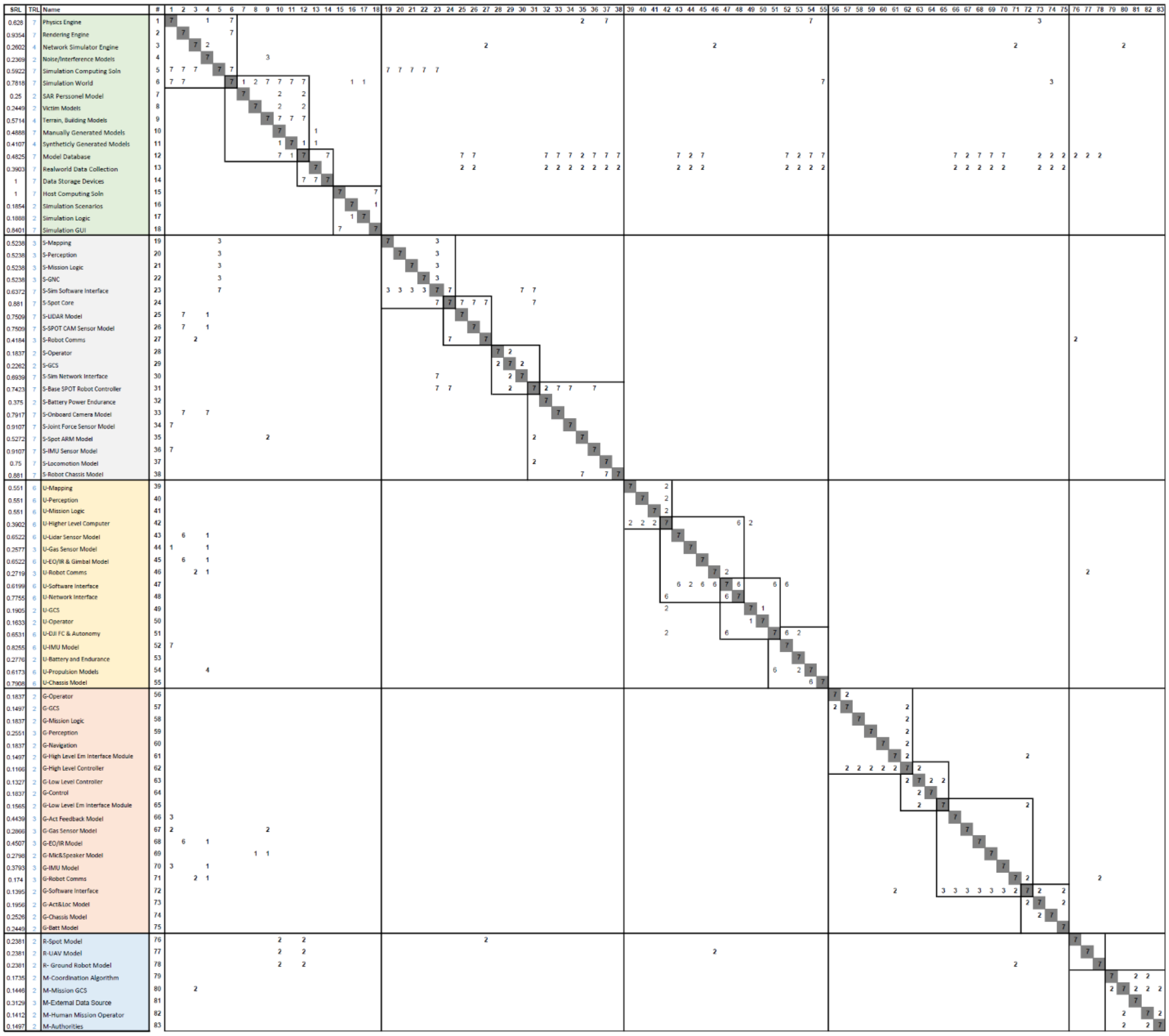
In general, the component readiness level of this ground robot simulation would be low as this is the most customized robot system compared to the previous two. The choice of the simulation environment is important to ensure that the relevant sensor models and mechanical simulation models are present. However, the simulation readiness can only be a maximum of 2 as the key functional requirements would be present in the simulation engine, but they cannot be tested until the robot models have been fully integrated hence allowing a proof of concept.

5.9.3.5 Assumptions Made for Multi-Robot Simulation

Similar to UGV Robotic Simulation, the multi-robot simulation has a lower readiness level as there can be no generic algorithms; this is currently an area of research and development, as seen from DARPA's efforts to organize competitions to foster growth in multi-robot collaboration for search and rescue. In general, since the robots have not been fully developed, the reduced order models of the robot behaviors could be challenging to develop.

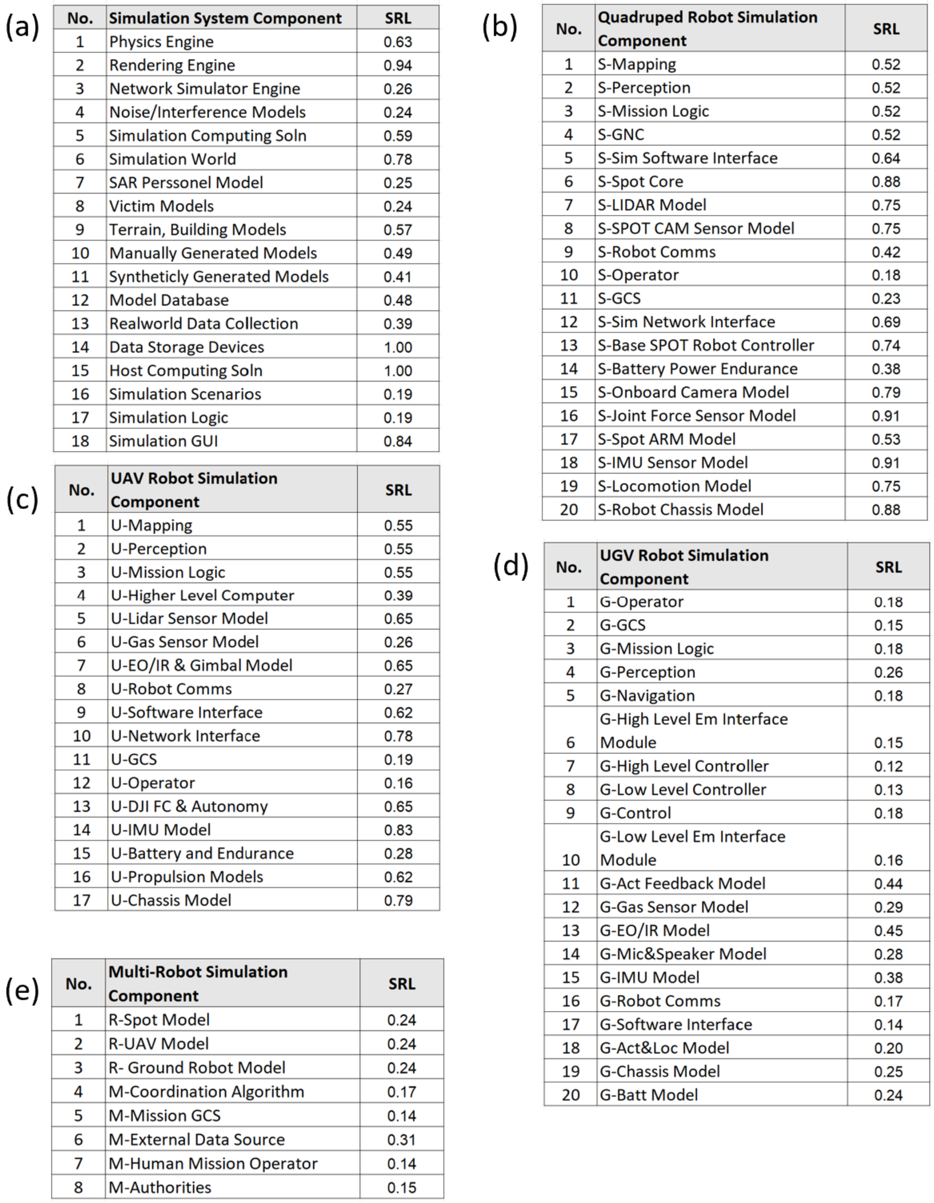
5.9.4 Calculating the SRL of the components

The TRL and IRL levels are populated, and the corresponding SRL is calculated. Figure 29 shows how the completed DSM table would resemble. Figure 30 shows the tables of the component SRL when represented in the five main groups. Due to space constraints, it will be difficult to read the entire DSM in Figure 29. Please refer to Appendix A for the TRL and IRL values used to populate the DSM shown in Figure 29, and the calculated SRL values will be found in the tables shown in Figure 30.



SRL	TRL	Name	#
0.628	7	Physics Engine	1
0.9354	7	Rendering Engine	2
0.2602	4	Network Simulator Engine	3
0.2369	2	Noise/Interference Models	4
0.5922	7	Simulation Computing Soln	5
0.7818	7	Simulation World	6
0.25	2	SAR Perssonel Model	7
0.2449	2	Victim Models	8
0.5714	4	Terrain, Building Models	9
0.4888	7	Manually Generated Models	10
0.4107	4	Syntheticly Generated Models	11
0.4825	7	Model Database	12
0.3903	7	Realworld Data Collection	13
1	7	Data Storage Devices	14
1	7	Host Computing Soln	15
0.1854	2	Simulation Scenarios	16
0.1888	2	Simulation Logic	17
0.8401	7	Simulation GUI	18
0.5238	3	S-Mapping	19
0.5238	3	S-Perception	20
0.5238	3	S-Mission Logic	21
0.5238	3	S-GNC	22
0.6372	7	S-Sim Software Interface	23
0.881	7	S-Spot Core	24
0.7509	7	S-LIDAR Model	25
0.7509	7	S-SPOT CAM Sensor Model	26
0.4184	3	S-Robot Comms	27
0.1837	2	S-Operator	28
0.2262	2	S-GCS	29
0.6939	7	S-Sim Network Interface	30
0.7423	7	S-Base SPOT Robot Controller	31
0.375	2	S-Battery Power Endurance	32
0.7917	7	S-Onboard Camera Model	33
0.9107	7	S-Joint Force Sensor Model	34
0.5272	7	S-Spot ARM Model	35
0.9107	7	S-IMU Sensor Model	36
0.75	7	S-Locomotion Model	37
0.881	7	S-Robot Chassis Model	38
0.551	6	U-Mapping	39
0.551	6	U-Perception	40
0.551	6	U-Mission Logic	41
0.3902	6	U-Higher Level Computer	42
0.6522	6	U-Lidar Sensor Model	43
0.2577	3	U-Gas Sensor Model	44
0.6522	6	U-EO/IR & Gimbal Model	45
0.2719	3	U-Robot Comms	46
0.6199	6	U-Software Interface	47
0.7755	6	U-Network Interface	48
0.1905	2	U-GCS	49
0.1633	2	U-Operator	50
0.6531	6	U-DJI FC & Autonomy	51
0.8255	6	U-IMU Model	52
0.2776	2	U-Battery and Endurance	53
0.6173	6	U-Propulsion Models	54
0.7908	6	U-Chassis Model	55
0.1837	2	G-Operator	56
0.1497	2	G-GCS	57
0.1837	2	G-Mission Logic	58
0.2551	3	G-Perception	59
0.1837	2	G-Navigation	60
0.1497	2	G-High Level Em Interface Module	61
0.1166	2	G-High Level Controller	62
0.1327	2	G-Low Level Controller	63
0.1837	2	G-Control	64
0.1565	2	G-Low Level Em Interface Module	65
0.4439	3	G-Act Feedback Model	66
0.2866	3	G-Gas Sensor Model	67
0.4507	3	G-EO/IR Model	68
0.2798	2	G-Mic&Speaker Model	69
0.3793	3	G-IMU Model	70
0.174	3	G-Robot Comms	71
0.1395	2	G-Software Interface	72
0.1956	2	G-Act&Loc Model	73
0.2526	2	G-Chassis Model	74
0.2449	2	G-Batt Model	75
0.2381	2	R-Spot Model	76
0.2381	2	R-UAV Model	77
0.2381	2	R- Ground Robot Model	78
0.1735	2	M-Coordination Algorithm	79
0.1446	2	M-Mission GCS	80
0.3129	3	M-External Data Source	81
0.1412	2	M-Human Mission Operator	82
0.1497	2	M-Authorities	83

Figure 29. DSM with IRL, TRL and calculated SRL



(a)

No.	Simulation System Component	SRL
1	Physics Engine	0.63
2	Rendering Engine	0.94
3	Network Simulator Engine	0.26
4	Noise/Interference Models	0.24
5	Simulation Computing Soln	0.59
6	Simulation World	0.78
7	SAR Perssonel Model	0.25
8	Victim Models	0.24
9	Terrain, Building Models	0.57
10	Manually Generated Models	0.49
11	Syntheticly Generated Models	0.41
12	Model Database	0.48
13	Realworld Data Collection	0.39
14	Data Storage Devices	1.00
15	Host Computing Soln	1.00
16	Simulation Scenarios	0.19
17	Simulation Logic	0.19
18	Simulation GUI	0.84

(b)

No.	Quadruped Robot Simulation Component	SRL
1	S-Mapping	0.52
2	S-Perception	0.52
3	S-Mission Logic	0.52
4	S-GNC	0.52
5	S-Sim Software Interface	0.64
6	S-Spot Core	0.88
7	S-LIDAR Model	0.75
8	S-SPOT CAM Sensor Model	0.75
9	S-Robot Comms	0.42
10	S-Operator	0.18
11	S-GCS	0.23
12	S-Sim Network Interface	0.69
13	S-Base SPOT Robot Controller	0.74
14	S-Battery Power Endurance	0.38
15	S-Onboard Camera Model	0.79
16	S-Joint Force Sensor Model	0.91
17	S-Spot ARM Model	0.53
18	S-IMU Sensor Model	0.91
19	S-Locomotion Model	0.75
20	S-Robot Chassis Model	0.88

(c)

No.	UAV Robot Simulation Component	SRL
1	U-Mapping	0.55
2	U-Perception	0.55
3	U-Mission Logic	0.55
4	U-Higher Level Computer	0.39
5	U-Lidar Sensor Model	0.65
6	U-Gas Sensor Model	0.26
7	U-EO/IR & Gimbal Model	0.65
8	U-Robot Comms	0.27
9	U-Software Interface	0.62
10	U-Network Interface	0.78
11	U-GCS	0.19
12	U-Operator	0.16
13	U-DJI FC & Autonomy	0.65
14	U-IMU Model	0.83
15	U-Battery and Endurance	0.28
16	U-Propulsion Models	0.62
17	U-Chassis Model	0.79

(d)

No.	UGV Robot Simulation Component	SRL
1	G-Operator	0.18
2	G-GCS	0.15
3	G-Mission Logic	0.18
4	G-Perception	0.26
5	G-Navigation	0.18
6	G-High Level Em Interface Module	0.15
7	G-High Level Controller	0.12
8	G-Low Level Controller	0.13
9	G-Control	0.18
10	G-Low Level Em Interface Module	0.16
11	G-Act Feedback Model	0.44
12	G-Gas Sensor Model	0.29
13	G-EO/IR Model	0.45
14	G-Mic&Speaker Model	0.28
15	G-IMU Model	0.38
16	G-Robot Comms	0.17
17	G-Software Interface	0.14
18	G-Act&Loc Model	0.20
19	G-Chassis Model	0.25
20	G-Batt Model	0.24

(e)

No.	Multi-Robot Simulation Component	SRL
1	R-Spot Model	0.24
2	R-UAV Model	0.24
3	R- Ground Robot Model	0.24
4	M-Coordination Algorithm	0.17
5	M-Mission GCS	0.14
6	M-External Data Source	0.31
7	M-Human Mission Operator	0.14
8	M-Authorities	0.15

Figure 30. Calculated System Readiness Levels of Components

5.9.4.1 Insights Gained from Calculated SRL

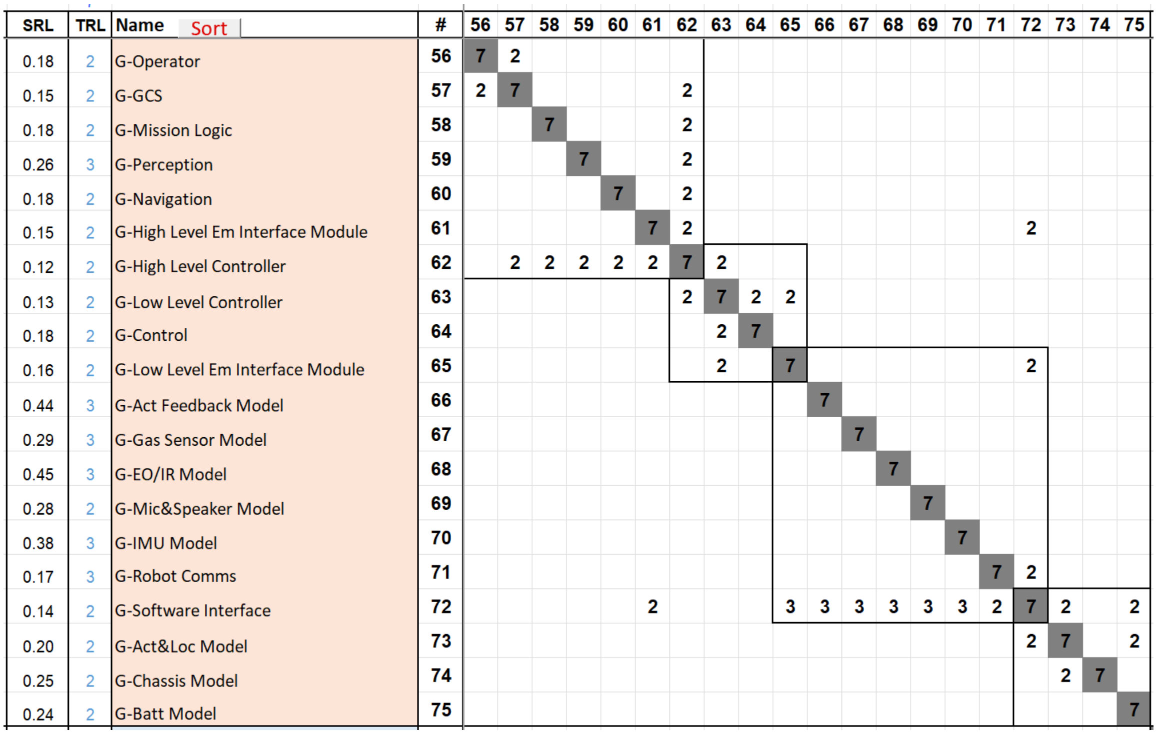
The following points highlight the insights gained from analyzing the SRL of the various groups of components.

<u>Ability to Track SRL during Simulation Capability Buildup.</u> In setting up the use-case in the chapter, the robotic systems used for this example were chosen according to the literature survey done and the maturity levels of the system and their respective availability of the simulation models. This would represent the starting state of the SRL for any organization that chooses these sets of robots and

simulation architecture, hence indicating the amount of effort needed by the organization to build up the simulation capabilities.

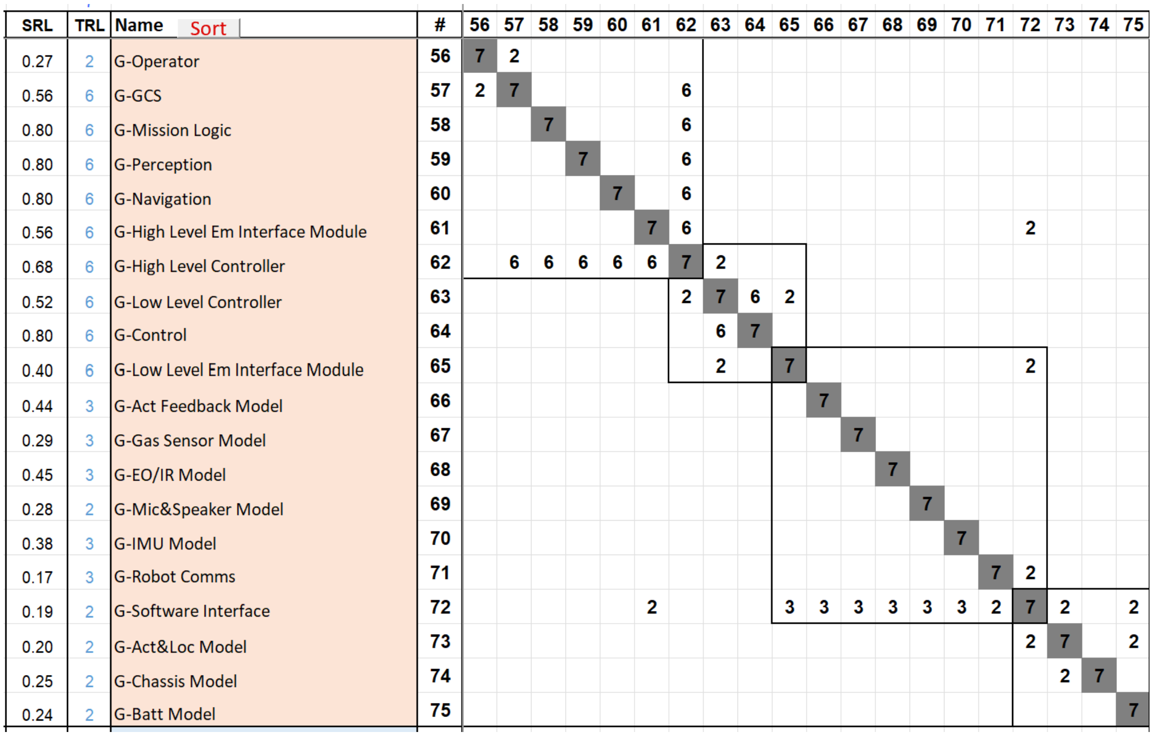
Using this method of the representation of SRL using the DSM would allow for the tracking of the development of key components and their readiness to be integrated into the robotic simulation setup. The DSM, together with the TRL and IRL models, is meant to be a live model to be updated as the progress of the robot and simulation development matures, hence allowing stakeholders to keep track of the overall readiness levels of key groups and components.

Using the build-up of the UGV simulation components as an example, a program manager will be able to request the individual engineers who are working on the different simulation components to update the status of the IRL and TRL based on the description given in **Error! Reference source not found.** and Table 7. The program manager will start off with the TRL and IRL values in Figure 31, resulting in the corresponding SRL values. Figure 32 shows the TRL and IRL values updated by the engineers after a period or at a critical milestone. For the first group of components, an engineer would have matured the interfaces between the components to IRL 6 and matured the components to TRL 6. Looking solely at the SRL indicators, we can see that there are several components that are not at the same SRL level as the other algorithm components. An obvious example to highlight would be the lagging GCS SRL is at 0.56, while the other algorithm components, such as mission logic, are at 0.80. This is a good indicator that components interfacing with the GCS are lacking in development. A quick look at the DSM would attribute the operator interface as the factor that is slowing down the GCS development; the operator interface likewise has a low SRL. However, looking within group 1 alone will not give any insights as to why the high-level controllers and interface modules have a lower SRL of 0.56 and 0.68. To understand why the interactions can be traced to the coupling of the low-level controller and interface module components found in group 2; these components have a low IRL of 2. Hence, the program manager or system engineer would be able to focus on facilitating the development of such interfaces.



SRL	TRL	Name Sort	#	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
0.18	2	G-Operator	56	7	2																		
0.15	2	G-GCS	57	2	7					2													
0.18	2	G-Mission Logic	58			7				2													
0.26	3	G-Perception	59				7			2													
0.18	2	G-Navigation	60					7		2													
0.15	2	G-High Level Em Interface Module	61						7	2										2			
0.12	2	G-High Level Controller	62		2	2	2	2	2	7	2												
0.13	2	G-Low Level Controller	63							2	7	2	2										
0.18	2	G-Control	64								2	7											
0.16	2	G-Low Level Em Interface Module	65								2		7							2			
0.44	3	G-Act Feedback Model	66											7									
0.29	3	G-Gas Sensor Model	67												7								
0.45	3	G-EO/IR Model	68													7							
0.28	2	G-Mic&Speaker Model	69														7						
0.38	3	G-IMU Model	70															7					
0.17	3	G-Robot Comms	71																7	2			
0.14	2	G-Software Interface	72						2				3	3	3	3	3	3	2	7	2		2
0.20	2	G-Act&Loc Model	73																	2	7		2
0.25	2	G-Chassis Model	74																		2	7	
0.24	2	G-Batt Model	75																				7

Figure 31. Start state of SRL for UGV Simulation



SRL	TRL	Name Sort	#	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
0.27	2	G-Operator	56	7	2																		
0.56	6	G-GCS	57	2	7					6													
0.80	6	G-Mission Logic	58			7				6													
0.80	6	G-Perception	59				7			6													
0.80	6	G-Navigation	60					7		6													
0.56	6	G-High Level Em Interface Module	61						7	6										2			
0.68	6	G-High Level Controller	62		6	6	6	6	6	7	2												
0.52	6	G-Low Level Controller	63							2	7	6	2										
0.80	6	G-Control	64								6	7											
0.40	6	G-Low Level Em Interface Module	65								2		7							2			
0.44	3	G-Act Feedback Model	66											7									
0.29	3	G-Gas Sensor Model	67												7								
0.45	3	G-EO/IR Model	68													7							
0.28	2	G-Mic&Speaker Model	69														7						
0.38	3	G-IMU Model	70															7					
0.17	3	G-Robot Comms	71																7	2			
0.19	2	G-Software Interface	72						2				3	3	3	3	3	3	2	7	2		2
0.20	2	G-Act&Loc Model	73																	2	7		2
0.25	2	G-Chassis Model	74																		2	7	
0.24	2	G-Batt Model	75																				7

Figure 32. Snapshot of present state of UGV Simulation

General Trend of Simulation Readiness of Various Systems. When comparing the various robotic systems, it can be noticed that the SPOT robotic system is the readiest system for simulation, while the UGV system is the least ready. This is in line with expectations as the SPOT system was used in the DARPA robotic challenge, and the models of the robots are available online as they were used for the simulation challenge [69], while there are software interfaces for the physical robot [68]. However, a fully customized robot would need extensive development before the robot can be fully simulated and if suitable, mainly the sensor models and the actuator models could be reused from other robotic simulations. Another general trend of the SRL metrics indicates that components that need to be developed using the simulator should have lower SRL values as these components need to be matured together with their interfaces.

The following points below highlights how the trend of the SRL of the individual components can be explained based on what was found in the literature research.

Trends Identified within Simulation System. Out of the three simulation engines needed, the network simulator is the least ready. This is since the chosen Gazebo simulator with ROS interface does not readily come with network simulation effects. With regards to the models to be used in the simulation, the SAR Personnel model is the least ready as

Trends Identified within SPOT Robot System. The least-ready components will be the human operator and the GCS, as these are components that need to be developed after the robot has been acquired. For example, the human operator will need to be trained on how to interface with the GCS, while the GCS would be a custom interface to match the configuration of the robot and the task at hand.

Trends Identified within UAV System. The overall trend of the Component SRLs is lower than that of the SPOT Robot as there are higher numbers of components that need to be developed in the UAV. However, the components that are already found on the base UAV should have a higher maturity level if a relevant simulation model can be found (e.g. flight controllers). Hence, the start state of the overall readiness level in adopting the UAV system is higher than the fully customized UGV system.

Trends Identified within UGV System. The UGV system has the lowest overall readiness level, as the robot will need to be built from the ground up. Hence, the interfaces would need to be built together with the subcomponents. For the simulation, various pre-existing models of common sensors or motors could be used to increase readiness levels.

Trends Identified within Multi-Robot System. The readiness of the multi-robot system for simulation would naturally be low due to two main reasons. First would be the fact that the multi-robot algorithms could be specific to the various types of robots used; it will not be at a high readiness level at the start of the project. The second is that the reduced-order models that would be needed to simulate high numbers of robot systems would not be available at the start of the project, and the behaviour of those models could be dependent on the outcome of the development of those systems. This is especially so if a higher fidelity behaviour of a reduced order model of the robotic systems is required. Hence, the overall readiness of this system is the lowest.

5.10 Conclusion

This chapter covered the application of the framework introduced in chapter 4. Using this framework, we can build a high-level multi-robot simulation architecture by incrementally building on the operational requirements of a USAR operation. This systematic approach to the build-up of the final simulation architecture will ensure that the simulation solution will meet the development requirements of the operational multi-robot system. The modelling of the architecture in the form of the DSM allows program managers or system engineers to better organize engineers who work building the simulators. With the aid of the DSM, the components of SRL were calculated. Managers will be able to use the SRL to appreciate the readiness level of the simulation system they intend to build, while the application of SRL at key points during their projects will allow the managers to have a snapshot of the current readiness, allowing them to make informed decisions along the way.

Figure 31 and Figure 32 represent a simple example of how managers are able to use the SRL to keep track of the development of simulation capabilities and identify possible components that are impeding the component's SRL. For complex cases consisting of a higher number of components and interactions, the components that are impeding the development and SRL may not be intuitive. The use of the DSMs and the SRL calculations will help the managers prioritize resources during the buildup of simulation capabilities and identify possible components that are impeding developments. This will allow the managers to re-prioritize these resources to ensure the component of interest meets its SRL target.

6 Discussion and Conclusion

6.1 Further Discussions on the Application of Framework

Integrating Digital Technologies to Boost SRL. The maximum allocated TRL and IRL levels used in this framework were 7. These values were allocated to the SPOT Quadruped Simulation model as the models were used in the DARPA Sub-T challenge, which consisted of both virtual and live demonstration cooperative heterogeneous robots. Hence, the simulated models were to be tested both live and virtually on a test scenario. In order to meet the TRL and IRL levels of 8 or 9, the system needs to be deployed in the real world. As the system is being tested on real-world scenarios, the simulation setup could benefit from the use of digital technology concepts, such as the creation of Digital Twins or Digital Shadows, to continuously improve the fidelity of the models used in the simulation. This feedback can be realized through the "Realworld Data Collection" component, which stores the collected data in the "Data Storage Devices" component. These components could be expanded within the simulation architecture to improve the resolution of the components needed to implement successful Digital Twins.

General Framework for Management of Multi-Robot Simulation Capability Development. This framework is currently applied to the field of USAR robotic operations. However, the steps found in this framework applied to any general multi-robotic system that has complex interactions. If the general requirements of the operations can be identified, various multi-robot simulation architectures can be built based on those requirements. The readiness for these simulation architectures based on the choice of the components used can then be explored, allowing decision-makers to make an informed choice about how their technical decisions impact the ability to build up the simulation capabilities needed for the simulation setup.

Comparing this Framework to Existing Simulation Frameworks. The framework presented in this thesis gives a proposal on how a high-level simulation architecture for a multi-robot system could be formulated and suggests a method of using SRL to allow managers to keep track of the development in the simulation capabilities. The proposed framework guides the user on how to generate a high-level simulation architecture and evaluate its SRL indicating the amount of effort needed to mature the simulation capabilities to a targeted SRL. The simulation frameworks that were found during literature survey does not provide such guidance and are targeted at engineers who needs to use a tool to immediately simulate robots for development. As noted by some of the frameworks highlighted in Chapter 2, different simulation technologies would need to be integrated together to suite a specific simulation purpose.

Hence, the work needs to be done to customize the readily available robotic simulation software to meet the needs of the robotic systems to be simulated. This makes it important for managers to track the readiness of a chosen simulation architecture based on the selection of simulation technologies and the chosen robotic platforms to be simulated. The proposed framework provides a useful tool for managers compared to the current existing simulation frameworks which do not provide insights on how to manage the development.

6.2 Limitations

This framework allows us to evaluate the readiness of the multi-robot simulation system based on the various technical decisions made, such as the choice of the robots, simulator or simulation engines, and software interfaces. However, it does not consider the specific development requirements of the robot simulation. Such requirements could be the ability to train images for perception or the need to have high-fidelity mechanical simulation.

The framework described in this thesis captures only the direct interface between components and does not consider how well two components will perform while working together. For example, the choice of a commercial gaming engine will yield better-simulated camera images that could be used for the training of perception algorithms. However, this benefit might come at the expense of having fewer built-in robot models that are ready to use, as the gaming engine needs to be modified for robot simulation. Conversely, a dedicated robot simulator may have more models supported and could be able to simulate mechanical interactions better. Hence, the SRL calculated using this framework will return a higher readiness level for the robot simulator compared to the gaming engine, even though the task at hand requires high-fidelity images.

To work around these limitations, managers or system engineers would need to calculate the SRL for both solutions and subsequently carry out a tradeoff to decide if it is beneficial to use a higher fidelity engine at the expense of more effort in building up the simulation setup.

6.3 Future Research

Based on the limitations described above, a recommendation of possible research could improve the resolution of information represented in the robot simulation architecture to include more types of interactions between the components. One example highlighted by Eppinger & Browning (2012) [51] demonstrates how the DSM could be used to represent three types of interactions; materials, energy and spatial adjacency. The DSM used in this framework could utilize another type of interaction that highlights how well-suited a component is to meet the objects of that component (e.g. a camera sensor could naturally require higher rendering fidelity engines). Adding new interactions would increase the overall complexity of the architectures represented in the diagram form. However, the added interactions could lead to new ways of clustering the DSM.

6.4 Conclusion

This thesis provided a framework that provides the user with a systematic process of building a high-level multi-robot simulation setup that can be used in the development of USAR robotic systems. The framework traces the high-level requirements needed from a USAR multi-robotic system concept starting from the USAR Operation, followed by the heterogeneous multi-robot system setup, the component robot system setup, and finally, the simulation architecture needed for both multi-robot and component robot simulations. The analysis tools proposed allows managers or decision-makers to estimate the technical readiness of the simulation architecture based on the selected simulation tools and robots chosen to be used in the system.

Bibliography

[1] A. Davids, “Urban search and rescue robots: from tragedy to technology,” *IEEE Intell. Syst.*, vol. 17, no. 2, pp. 81–83, Mar. 2002, doi: 10.1109/MIS.2002.999224.

[2] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Y. Nakagawa, “Test Arenas and Performance Metrics for Urban Search and Rescue Robots,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, Las Vegas, NV, USA, 2003, vol. 3, pp. 3396–3403. doi: 10.1109/IROS.2003.1249681.

[3] J. Collins, S. Chand, A. Vanderkop, and D. Howard, “A Review of Physics Simulators for Robotic Applications,” *IEEE Access*, vol. 9, pp. 51416–51431, 2021, doi: 10.1109/ACCESS.2021.3068769.

[4] M. Santos Pessoa de Melo, J. Gomes da Silva Neto, P. Jorge Lima da Silva, J. M. X. Natario Teixeira, and V. Teichrieb, “Analysis and Comparison of Robotics 3D Simulators,” in *2019 21st Symposium on Virtual and Augmented Reality (SVR)*, Rio de Janeiro, Brazil, Oct. 2019, pp. 242–251. doi: 10.1109/SVR.2019.00049.

[5] J. P. Queralta *et al.*, “Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision,” *IEEE Access*, vol. 8, pp. 191617–191643, 2020, doi: 10.1109/ACCESS.2020.3030190.

[6] V. Futch and A. Allen, “Search and Rescue Applications: On the Need to Improve Ocean Observing Data Systems in Offshore or Remote Locations,” *Front. Mar. Sci.*, vol. 6, p. 301, Jun. 2019, doi: 10.3389/fmars.2019.00301.

[7] J. P. Queralta, J. Raitoharju, T. N. Gia, N. Passalis, and T. Westerlund, “AutoSOS: Towards Multi-UAV Systems Supporting Maritime Search and Rescue with Lightweight AI and Edge Computing.” arXiv, May 07, 2020. Accessed: Jan. 03, 2023. [Online]. Available: http://arxiv.org/abs/2005.03409

[8] A. Matos *et al.*, “Multiple robot operations for maritime search and rescue in euRathlon 2015 competition,” in *OCEANS 2016 - Shanghai*, Shanghai, Apr. 2016, pp. 1–7. doi: 10.1109/OCEANSAP.2016.7485707.

[9] P. Petracek, V. Kratky, M. Petrlik, T. Baca, R. Kratochvil, and M. Saska, “Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7596–7603, Oct. 2021, doi: 10.1109/LRA.2021.3098304.

[10] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose UAV for search and rescue operations in mountain avalanche events,” *Geomat. Nat. Hazards Risk*, vol. 8, no. 1, pp. 18–33, Jan. 2017, doi: 10.1080/19475705.2016.1238852.

[11] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, “A Multirobot Path-Planning Strategy for Autonomous Wilderness Search and Rescue,” *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015, doi: 10.1109/TCYB.2014.2360368.

[12] D. Greer and P. McKerrow, “Robots in Urban Search and Rescue Operations,” *ARAA*, 2002.

[13] R. R. Murphy, J. Casper, and M. Micire, “Potential Tasks and Research Issues for Mobile Robots in RoboCup Rescue,” in *RoboCup 2000: Robot Soccer World Cup IV*, vol. 2019, P. Stone, T. Balch, and G. Kraetzschmar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 339–344. doi: 10.1007/3-540-45324-5_36.

[14] R. R. Murphy, “Activities of the rescue robots at the World Trade Center from 11-21 september 2001,” *IEEE Robot. Autom. Mag.*, vol. 11, no. 3, pp. 50–61, Sep. 2004, doi: 10.1109/MRA.2004.1337826.

[15] L. Chen and E. Miller-Hooks, “Optimal team deployment in urban search and rescue,” *Transp. Res. Part B Methodol.*, vol. 46, no. 8, pp. 984–999, Sep. 2012, doi: 10.1016/j.trb.2012.03.004.

[16] C. Chang and R. R. Murphy, “Towards Robot-Assisted Mass-Casualty Triage,” in *2007 IEEE International Conference on Networking, Sensing and Control*, London, UK, 2007, pp. 267–272. doi: 10.1109/ICNSC.2007.372789.

[17] Y. Liu and G. Nejat, “Robotic Urban Search and Rescue: A Survey from the Control Perspective,” *J. Intell. Robot. Syst.*, vol. 72, no. 2, pp. 147–165, Nov. 2013, doi: 10.1007/s10846-013-9822-x.

[18] C. Armbrust, G. D. Cubber, and K. Berns, “ICARUS - Control Systems for Search and Rescue Robots,” p. 17.

[19] M. Baker, R. Casey, B. Keyes, and H. A. Yanco, “Improved interfaces for human-robot interaction in urban search and rescue,” in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, The Hague, Netherlands, 2004, vol. 3, pp. 2960–2965. doi: 10.1109/ICSMC.2004.1400783.

[20] D. Doroftei, A. Matos, and G. de Cubber, “Designing Search and Rescue Robots towards Realistic User Requirements,” *Appl. Mech. Mater.*, vol. 658, pp. 612–617, Oct. 2014, doi: 10.4028/www.scientific.net/AMM.658.612.

[21] K. Oishi *et al.*, “A Hybrid Architecture for USAR Robot Development and Human User Testing,” p. 8, Oct. 2011.

[22] M. W. Kadous, R. K.-M. Sheh, and C. Sammut, “CASTER: A Robot for Urban Search and Rescue,” p. 11, Mar. 2012.

[23] H. Choi *et al.*, “On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward,” *Proc. Natl. Acad. Sci.*, vol. 118, no. 1, p. e1907856118, Jan. 2021, doi: 10.1073/pnas.1907856118.

[24] R. Ramin Shamshiri *et al.*, “Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison,” *Int. J. Agric. Biol. Eng.*, vol. 11, no. 4, pp. 12–20, 2018, doi: 10.25165/j.ijabe.20181104.4032.

[25] M. L. Loper and C. Turnitsa, “History of Combat Modeling and Distributed Simulation,” in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012, pp. 329–355. doi: 10.1002/9781118180310.ch16.

[26] R. R. Hill, A. Tolk, D. D. Hodson, and J. R. Millar, “OPEN CHALLENGES IN BUILDING COMBAT SIMULATION SYSTEMS TO SUPPORT TEST, ANALYSIS AND TRAINING,” in *2018 Winter Simulation Conference (WSC)*, Gothenburg, Sweden, Dec. 2018, pp. 3730–3741. doi: 10.1109/WSC.2018.8632233.

[27] N. Hooshangi and A. Alesheikh, “Developing an Agent-Based Simulation System for Post-Earthquake Operations in Uncertainty Conditions: A Proposed Method for Collaboration among Agents,” *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 1, p. 27, Jan. 2018, doi: 10.3390/ijgi7010027.

[28] N. Hooshangi, A. A. Alesheikh, M. Panahi, and S. Lee, “USAR simulation system: presenting spatial strategies in agents’task allocation under uncertainties,” Earthquake Hazards, preprint, Nov. 2020. doi: 10.5194/nhess-2020-277.

[29] K. Aloui, M. Hammadi, A. Guizani, M. Haddar, and T. Soriano, “A new SysML Model for UAV Swarm Modeling: UavSwarmML,” in *2022 IEEE International Systems Conference (SysCon)*, Montreal, QC, Canada, Apr. 2022, pp. 1–8. doi: 10.1109/SysCon53536.2022.9773922.

[30] M. A. Luna, A. Refaat Ragab, M. S. Ale Isac, P. Flores Pena, and P. C. Cervera, “A New Algorithm Using Hybrid UAV Swarm Control System for Firefighting Dynamical Task Allocation,” in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Melbourne, Australia, Oct. 2021, pp. 655–660. doi: 10.1109/SMC52423.2021.9659275.

[31] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch, “The EU-ICARUS project: Developing assistive robotic tools for search and rescue operations,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linkoping, Sweden, Oct. 2013, pp. 1–4. doi: 10.1109/SSRR.2013.6719323.

[32] T. Shih and H. Chang, “FPGA based hardware in the loop test platform of small size UAV,” in *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*, Daejeon, Korea (South), Dec. 2009, pp. 551–556. doi: 10.1109/CIRA.2009.5423228.

[33] L. Yuan *et al.*, "Modeling and Simulation of Stress and Strain of Chassis Structure for Wheeled Robot," *DEStech Trans. Comput. Sci. Eng.*, no. amms, Nov. 2018, doi: 10.12783/dtcse/amms2018/26221.

[34] D. M. Lewis, J. Bommer, G. E. Hindman, and S. F. Gregson, "Traditional to Modern Antenna Test Environments: The Impact of Robotics And Computational Electromagnetic Simulation on Modern Antenna Measurements," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, Dusseldorf, Germany, Mar. 2021, pp. 1–5. doi: 10.23919/EuCAP51087.2021.9411068.

[35] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research," *Sensors*, vol. 19, no. 3, p. 648, Feb. 2019, doi: 10.3390/s19030648.

[36] M. Kudelski, M. Cinus, L. Gambardella, and G. A. Di Caro, "A framework for realistic simulation of networked multi-robot systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, Oct. 2012, pp. 5018–5025. doi: 10.1109/IROS.2012.6385998.

[37] M. Patchou, C. Arendt, P. Gorczak, J. Guldenring, J. Tiemann, and C. Wietfeld, "Hardware in the Simulation Loop Framework For Reproducible Testing of Rescue Robot Communications in Constrained Environments," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Abu Dhabi, United Arab Emirates, Nov. 2020, pp. 348–353. doi: 10.1109/SSRR50563.2020.9292584.

[38] H. Han *et al.*, "Networked Robotic System Simulation Platform: Hardware-in-the-Loop Architecture, Enabling Technologies and Implementations," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, Oct. 2020, pp. 1194–1199. doi: 10.1109/WCSP49889.2020.9299745.

[39] S. Park, W. G. La, W. Lee, and H. Kim, "Devising a Distributed Co-Simulator for a Multi-UAV Network," *Sensors*, vol. 20, no. 21, p. 6196, Oct. 2020, doi: 10.3390/s20216196.

[40] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "USARSim: a robot simulator for research and education," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 1400–1405. doi: 10.1109/ROBOT.2007.363180.

[41] M. Lewis, J. Wang, and S. Hughes, "USARSim: Simulation for the Study of Human-Robot Interaction," *J. Cogn. Eng. Decis. Mak.*, vol. 1, no. 1, pp. 98–120, Mar. 2007, doi: 10.1177/155534340700100105.

[42] A. Harris and J. M. Conrad, "Survey of popular robotics simulators, frameworks, and toolkits," in *2011 Proceedings of IEEE Southeastcon*, Nashville, TN, USA, Mar. 2011, pp. 243–249. doi: 10.1109/SECON.2011.5752942.

[43] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, vol. 5, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 621–635. doi: 10.1007/978-3-319-67361-5_40.

[44] J. Lima, P. Costa, T. Brito, and L. Piardi, "Hardware-in-the-loop simulation approach for the Robot at Factory Lite competition proposal," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Porto, Portugal, Apr. 2019, pp. 1–6. doi: 10.1109/ICARSC.2019.8733649.

[45] I. Tejado, J. Serrano, E. Pérez, D. Torres, and B. M. Vinagre, "Low-cost Hardware-in-the-loop Testbed of a Mobile Robot to Support Learning in Automatic Control and Robotics**This work has been supported by the Spanish Ministry of Economy and Competitiveness under the project DPI2012-37062-C02-02 and the Junta de Extremadura under the Ayuda a Grupos with reference GR15178.," *IFAC-Pap.*, vol. 49, no. 6, pp. 242–247, 2016, doi: 10.1016/j.ifacol.2016.07.184.

[46] A. Hussein and H. Abbass, "Mixed Initiative Systems for Human-Swarm Interaction: Opportunities and Challenges." arXiv, Aug. 19, 2018. Accessed: Jan. 11, 2023. [Online]. Available: http://arxiv.org/abs/1808.06211

[47] M. Lyssenko, C. Gladisch, C. Heinzemann, M. Woehrle, and R. Triebel, "Instance Segmentation in CARLA: Methodology and Analysis for Pedestrian-oriented Synthetic Data Generation in Crowded

Scenes," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, Oct. 2021, pp. 988–996. doi: 10.1109/ICCVW54120.2021.00115.

[48] B. Birkmire, "The Advanced Framework for Simulation, Integration, and Modeling (AFSIM)," p. 11, Apr. 2021.

[49] S. Böttger, T.-C. Çallar, A. Schweikard, and E. Rückert, "Medical robotics simulation framework for application-specific optimal kinematics," *Curr. Dir. Biomed. Eng.*, vol. 5, no. 1, pp. 145–148, Sep. 2019, doi: 10.1515/cdbme-2019-0037.

[50] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning." arXiv, Sep. 25, 2020. Accessed: Sep. 02, 2022. [Online]. Available: http://arxiv.org/abs/2009.12293

[51] S. D. Eppinger and T. R. Browning, *Design structure matrix methods and applications*. Cambridge, Mass: MIT Press, 2012.

[52] "Cambridge Advanced Modeller 2." https://camtoolkit.eng.cam.ac.uk/ (accessed Jan. 04, 2023).

[53] M. F. Austin and D. M. York, "System Readiness Assessment (SRA) an Illustrative Example," *Procedia Comput. Sci.*, vol. 44, pp. 486–496, 2015, doi: 10.1016/j.procs.2015.03.031.

[54] J. C. Mankins, "TECHNOLOGY READINESS LEVELS," *Off. Space Access Technol. NASA*, Apr. 1995.

[55] R. Clay, M. Shneider, S. Marburger, and T. Trucano, "Modeling and simulation technology readiness levels.," SAND2007-0570, 901709, Jan. 2006. doi: 10.2172/901709.

[56] F. Goolsby, "System of Systems Composition aNd Course of Action Pathfinding Tool (CNCPT)," Thesis, Massachusetts Institute of Technology, 2021.

[57] E. Cayirci, R. AlNaimi, and S. S. AlNabet, "Computer Assisted Military Experimentations," in *2022 Winter Simulation Conference (WSC)*, Singapore, Dec. 2022, pp. 1311–1324. doi: 10.1109/WSC57314.2022.10015294.

[58] B. Balaguer, S. Balakirsky, S. Carpin, M. Lewis, and C. Scrapper, "USARSim: a validated simulator for research in robotics and automation," *Workshop Robot Simulators Available Softw. Sci. Appl. Future Trends IEEERSJ*, p. 6, 2008.

[59] T. A. J. Schoonderwoerd, E. M. van Zoelen, K. van den Bosch, and M. A. Neerincx, "Design patterns for human-AI co-learning: A wizard-of-Oz evaluation in an urban-search-and-rescue task," *Int. J. Hum.-Comput. Stud.*, vol. 164, p. 102831, Aug. 2022, doi: 10.1016/j.ijhcs.2022.102831.

[60] A. Torres-Pardo *et al.*, "Legged locomotion over irregular terrains: state of the art of human and robot performance," *Bioinspir. Biomim.*, vol. 17, no. 6, p. 061002, Nov. 2022, doi: 10.1088/1748-3190/ac92b3.

[61] "Spot 2.0 from Boston Dynamics features improved autonomy and mobility." https://www.therobotreport.com/spot-20-launches-with-improved-autonomy-mobility/ (accessed Jan. 26, 2023).

[62] X. Xiao and R. Murphy, "A review on snake robot testbeds in granular and restricted maneuverability spaces," *Robot. Auton. Syst.*, vol. 110, pp. 160–172, Dec. 2018, doi: 10.1016/j.robot.2018.10.003.

[63] A. Shahoud, D. Shashev, and S. Shidlovskiy, "Mutual-Aided INS/Vision Navigation System Analysis and Optimization Using Sequential Filtering with State Recalculation," *Sensors*, vol. 23, no. 1, p. 79, Dec. 2022, doi: 10.3390/s23010079.

[64] "Snakes and Snake-like Robots Show How Sidewinders Conquer Sandy Slopes." https://www.cmu.edu/news/stories/archives/2014/october/october9_snakesconquerslopes.html (accessed Jan. 29, 2023).

[65] "SARCOS Guardian-S Magnetic Robotic Crawler System – Pine Environmental." https://www.pine-environmental.com/products/sarcos-guardian-s-magnetic-robotic-crawler-system (accessed Jan. 29, 2023).

[66] M. Zaratti, M. Fratarcangeli, and L. Iocchi, "A 3D Simulator of Multiple Legged Robots Based on USARSim," in *RoboCup 2006: Robot Soccer World Cup X*, vol. 4434, G. Lakemeyer, E. Sklar, D. G.

Sorrenti, and T. Takahashi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 13–24. doi: 10.1007/978-3-540-74024-7_2.

[67] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A Modular Gazebo MAV Simulator Framework," in *Robot Operating System (ROS)*, vol. 625, A. Koubaa, Ed. Cham: Springer International Publishing, 2016, pp. 595–625. doi: 10.1007/978-3-319-26054-9_23.

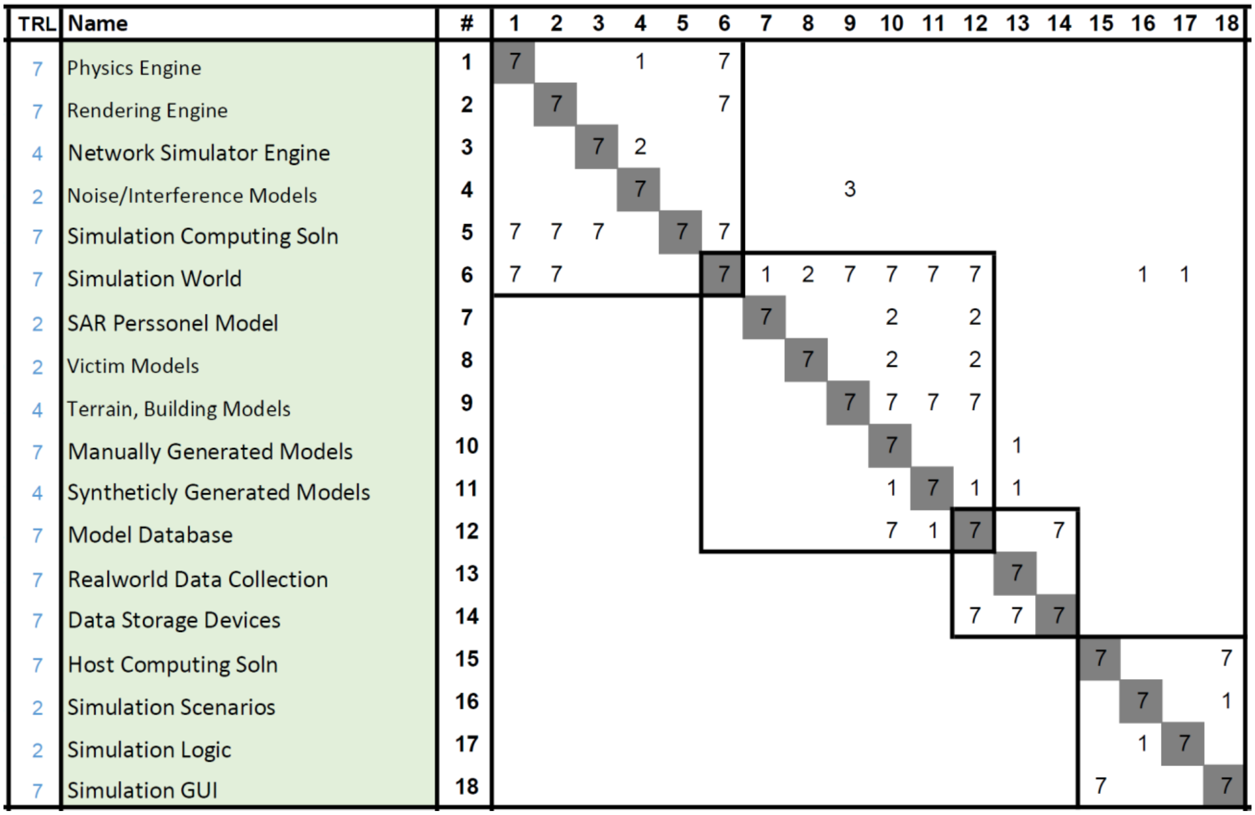
[68] "Robots/Spot - ROS Wiki." http://wiki.ros.org/Robots/Spot (accessed Jan. 04, 2023).

[69] "Gazebo." https://app.gazebosim.org (accessed Jan. 04, 2023).

Appendix A

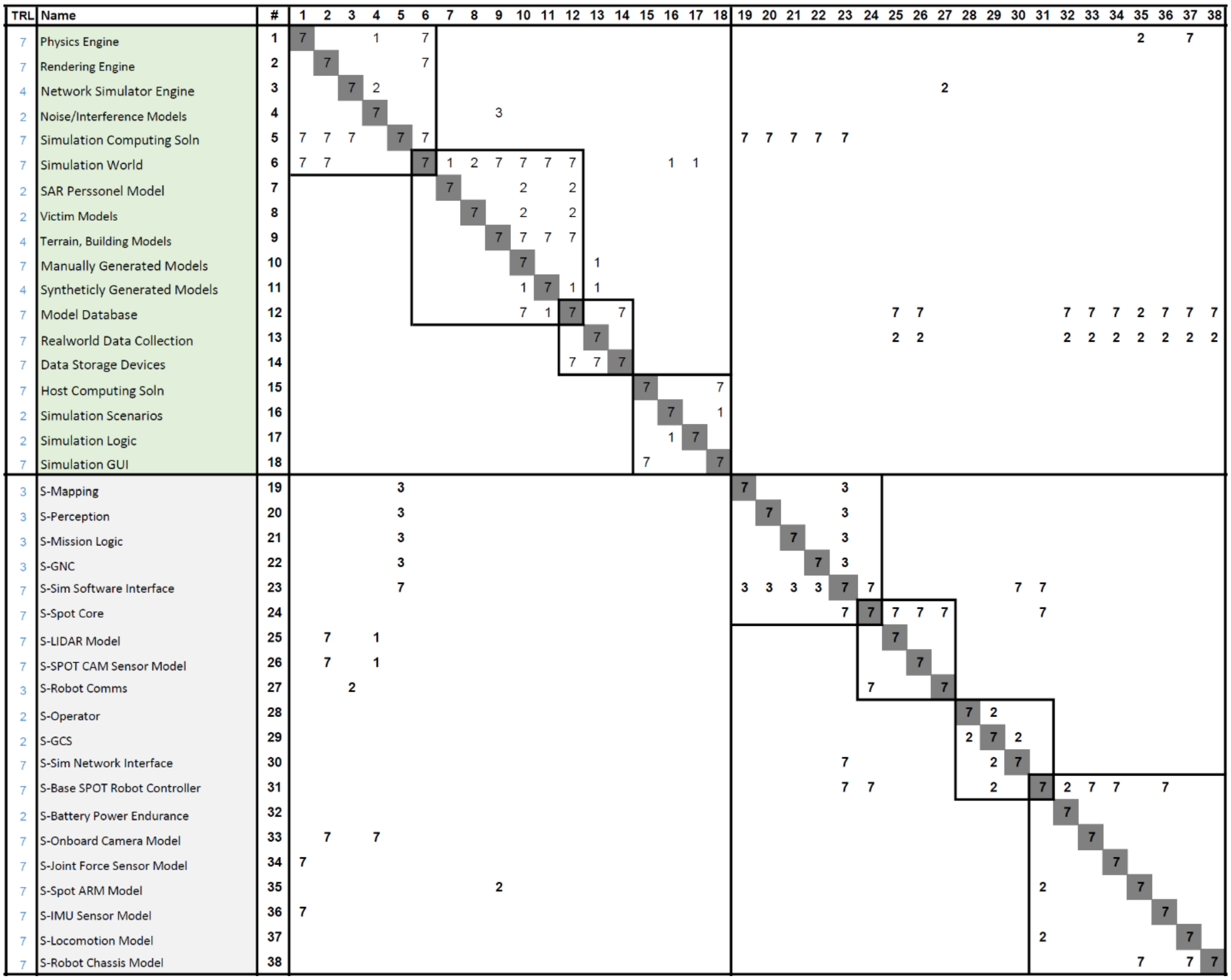
List of Simulation Components and Corresponding TRL and IRL levels used in Chapter 5

Add in the values that I used to build the figure.



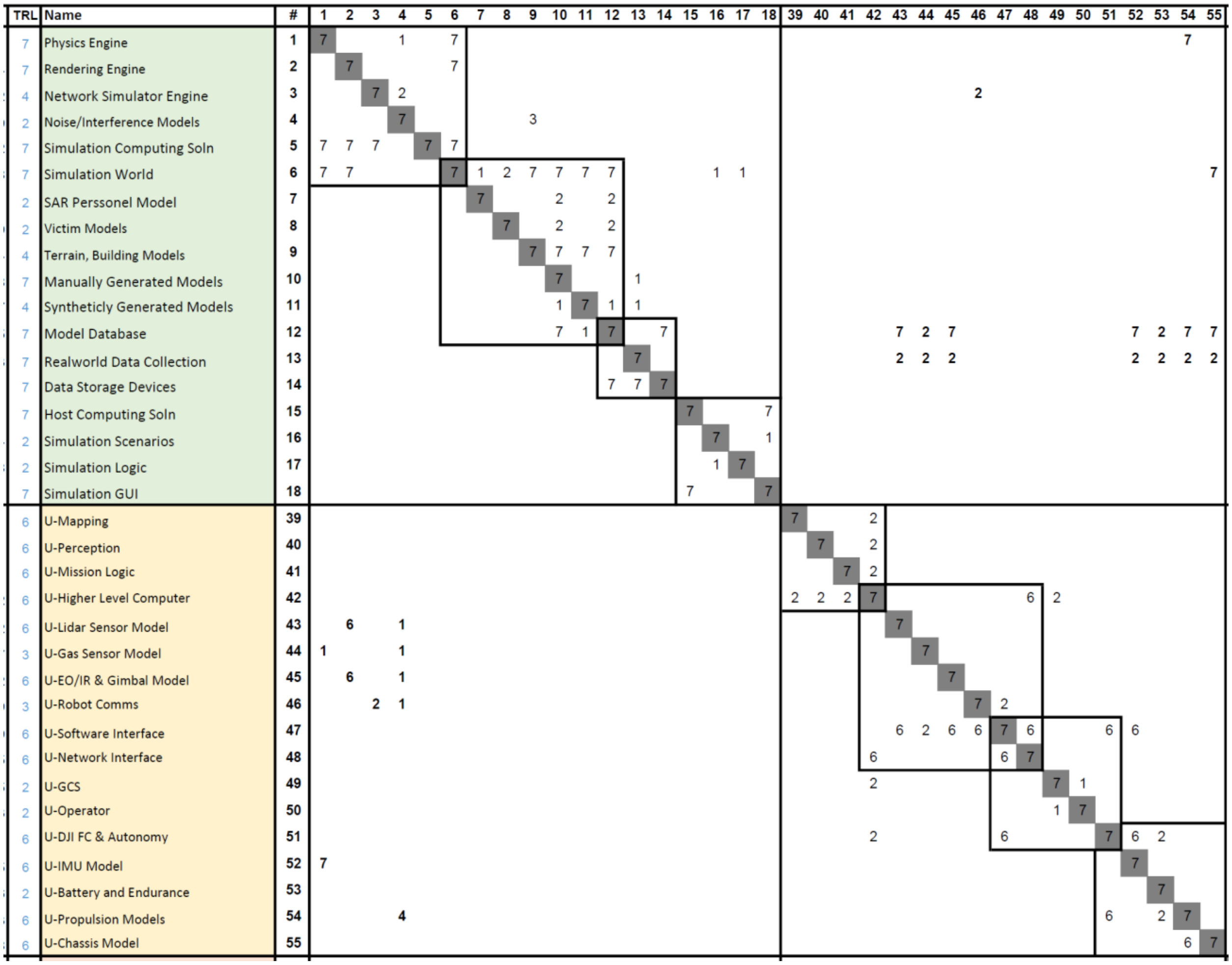
TRL	Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
7	Physics Engine	1	7			1		7												
7	Rendering Engine	2		7				7												
4	Network Simulator Engine	3			7	2														
2	Noise/Interference Models	4				7					3									
7	Simulation Computing Soln	5	7	7	7		7	7												
7	Simulation World	6	7	7				7	1	2	7	7	7	7				1	1	
2	SAR Perssonel Model	7							7			2		2						
2	Victim Models	8								7		2		2						
4	Terrain, Building Models	9									7	7	7	7						
7	Manually Generated Models	10										7			1					
4	Syntheticly Generated Models	11										1	7	1	1					
7	Model Database	12										7	1	7		7				
7	Realworld Data Collection	13													7					
7	Data Storage Devices	14												7	7	7				
7	Host Computing Soln	15															7			7
2	Simulation Scenarios	16																7		1
2	Simulation Logic	17																1	7	
7	Simulation GUI	18															7			7

Figure 33. Simulation System Components with TRL and IRL Assignments



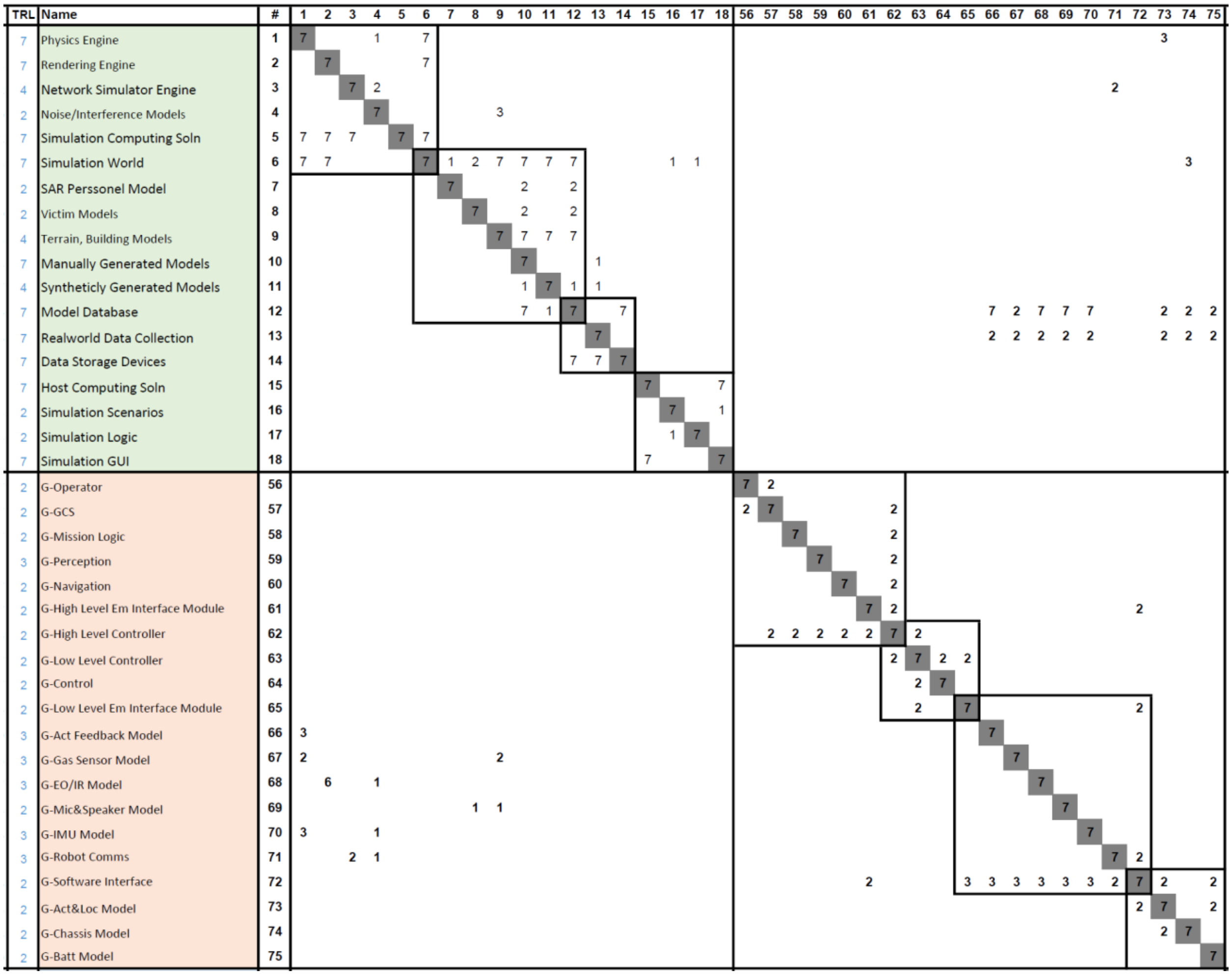
TRL	Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
7	Physics Engine	1	7			1		7																													2		7	
7	Rendering Engine	2		7				7																																
4	Network Simulator Engine	3			7	2																							2											
2	Noise/Interference Models	4				7					3																													
7	Simulation Computing Soln	5	7	7	7		7	7													7	7	7	7	7															
7	Simulation World	6	7	7				7	1	2	7	7	7	7				1	1																					
2	SAR Perssonel Model	7							7			2		2																										
2	Victim Models	8								7		2		2																										
4	Terrain, Building Models	9									7	7	7	7																										
7	Manually Generated Models	10										7			1																									
4	Syntheticly Generated Models	11										1	7	1	1																									
7	Model Database	12										7	1	7		7											7	7						7	7	7	2	7	7	7
7	Realworld Data Collection	13													7												2	2						2	2	2	2	2	2	2
7	Data Storage Devices	14												7	7	7																								
7	Host Computing Soln	15															7			7																				
2	Simulation Scenarios	16																7		1																				
2	Simulation Logic	17																1	7																					
7	Simulation GUI	18															7			7																				
3	S-Mapping	19					3														7				3															
3	S-Perception	20					3															7			3															
3	S-Mission Logic	21					3																7		3															
3	S-GNC	22					3																	7	3															
7	S-Sim Software Interface	23					7														3	3	3	3	7	7						7	7							
7	S-Spot Core	24																							7	7	7	7	7				7							
7	S-LIDAR Model	25		7		1																					7													
7	S-SPOT CAM Sensor Model	26		7		1																						7												
3	S-Robot Comms	27			2																					7			7											
2	S-Operator	28																												7	2									
2	S-GCS	29																												2	7	2								
7	S-Sim Network Interface	30																							7						2	7								
7	S-Base SPOT Robot Controller	31																							7	7					2		7	2	7	7		7		
2	S-Battery Power Endurance	32																																7						
7	S-Onboard Camera Model	33		7		7																													7					
7	S-Joint Force Sensor Model	34	7																																	7				
7	S-Spot ARM Model	35									2																						2				7			
7	S-IMU Sensor Model	36	7																																			7		
7	S-Locomotion Model	37																															2						7	
7	S-Robot Chassis Model	38																																			7		7	7

Figure 34. Quadruped Robot Simulation Components with TRL and IRL Assignments



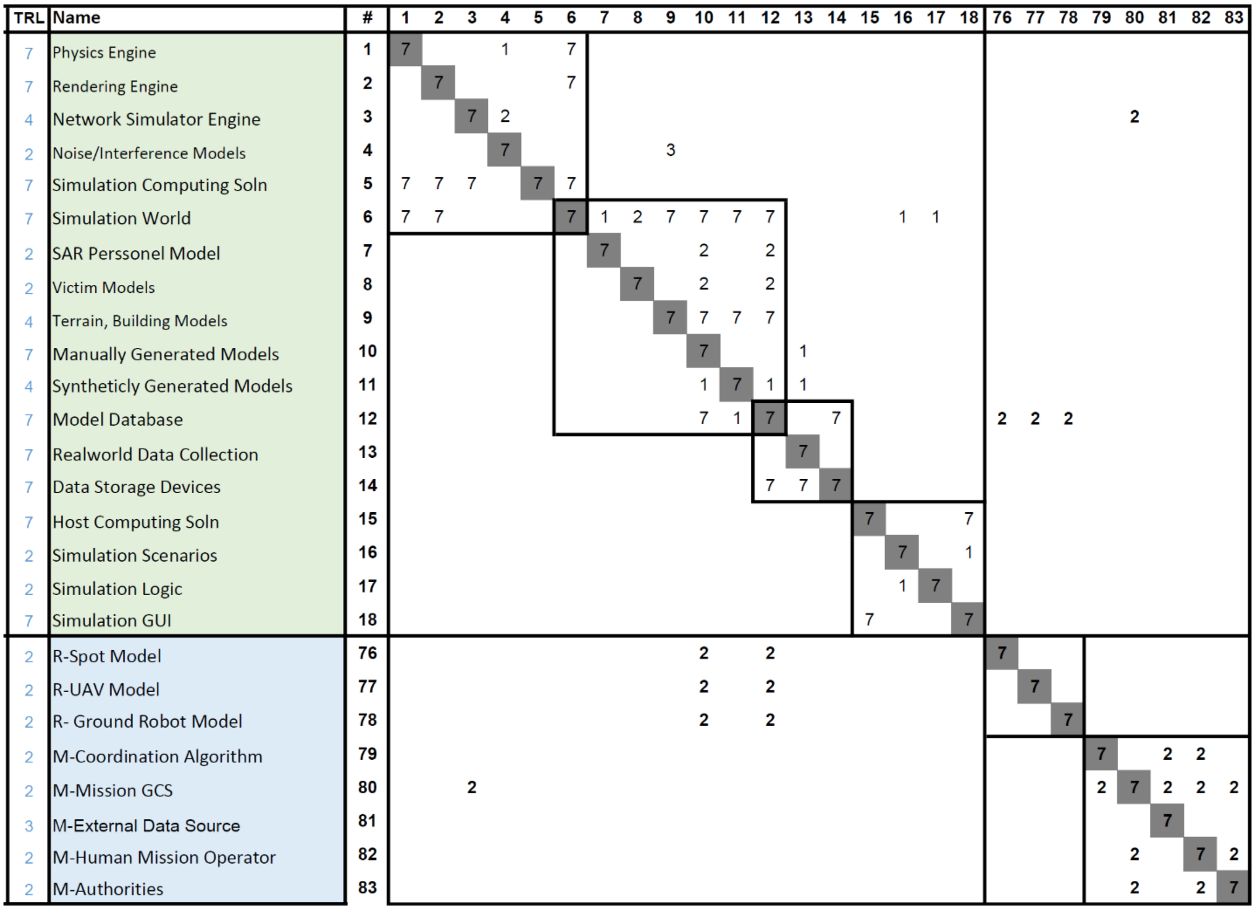
TRL	Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
7	Physics Engine	1	7			1		7																												7	
7	Rendering Engine	2		7				7																													
4	Network Simulator Engine	3			7	2																						2									
2	Noise/Interference Models	4				7					3																										
7	Simulation Computing Soln	5	7	7	7		7	7																													
7	Simulation World	6	7	7				7	1	2	7	7	7	7				1	1																		7
2	SAR Perssonel Model	7							7			2		2																							
2	Victim Models	8								7		2		2																							
4	Terrain, Building Models	9									7	7	7	7																							
7	Manually Generated Models	10										7			1																						
4	Syntheticly Generated Models	11										1	7	1	1																						
7	Model Database	12										7	1	7		7									7	2	7							7	2	7	7
7	Realworld Data Collection	13													7										2	2	2							2	2	2	2
7	Data Storage Devices	14												7	7	7																					
7	Host Computing Soln	15															7			7																	
2	Simulation Scenarios	16																7		1																	
2	Simulation Logic	17																1	7																		
7	Simulation GUI	18															7			7																	
6	U-Mapping	39																			7			2													
6	U-Perception	40																				7		2													
6	U-Mission Logic	41																					7	2													
6	U-Higher Level Computer	42																			2	2	2	7						6	2						
6	U-Lidar Sensor Model	43		6		1																			7												
3	U-Gas Sensor Model	44	1			1																				7											
6	U-EO/IR & Gimbal Model	45		6		1																					7										
3	U-Robot Comms	46			2	1																						7	2								
6	U-Software Interface	47																							6	2	6	6	7	6			6	6			
6	U-Network Interface	48																						6					6	7							
2	U-GCS	49																						2							7	1					
2	U-Operator	50																													1	7					
6	U-DJI FC & Autonomy	51																						2					6				7	6	2		
6	U-IMU Model	52	7																															7			
2	U-Battery and Endurance	53																																	7		
6	U-Propulsion Models	54				4																											6		2	7	
6	U-Chassis Model	55																																		6	7

Figure 35. UAV Robot Simulation Components with TRL and IRL Assignments



TRL	Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
7	Physics Engine	1	7			1		7																														3		
7	Rendering Engine	2		7				7																																
4	Network Simulator Engine	3			7	2																														2				
2	Noise/Interference Models	4				7					3																													
7	Simulation Computing Soln	5	7	7	7		7	7																																
7	Simulation World	6	7	7				7	1	2	7	7	7	7				1	1																				3	
2	SAR Perssonel Model	7							7			2		2																										
2	Victim Models	8								7		2		2																										
4	Terrain, Building Models	9									7	7	7	7																										
7	Manually Generated Models	10										7			1																									
4	Syntheticly Generated Models	11										1	7	1	1																									
7	Model Database	12										7	1	7		7																7	2	7	7	7			2	2	2
7	Realworld Data Collection	13													7																	2	2	2	2	2			2	2	2
7	Data Storage Devices	14												7	7	7																								
7	Host Computing Soln	15															7			7																				
2	Simulation Scenarios	16																7		1																				
2	Simulation Logic	17																1	7																					
7	Simulation GUI	18															7			7																				
2	G-Operator	56																			7	2																		
2	G-GCS	57																			2	7					2													
2	G-Mission Logic	58																					7				2													
3	G-Perception	59																						7			2													
2	G-Navigation	60																							7		2													
2	G-High Level Em Interface Module	61																								7	2										2			
2	G-High Level Controller	62																				2	2	2	2	2	7	2												
2	G-Low Level Controller	63																									2	7	2	2										
2	G-Control	64																										2	7											
2	G-Low Level Em Interface Module	65																										2		7							2			
3	G-Act Feedback Model	66	3																												7									
3	G-Gas Sensor Model	67	2								2																					7								
3	G-EO/IR Model	68		6		1																											7							
2	G-Mic&Speaker Model	69								1	1																							7						
3	G-IMU Model	70	3			1																													7					
3	G-Robot Comms	71			2	1																														7	2			
2	G-Software Interface	72																								2				3	3	3	3	3	3	2	7	2		2
2	G-Act&Loc Model	73																																			2	7		2
2	G-Chassis Model	74																																				2	7	
2	G-Batt Model	75																																						7

Figure 36. UGV Robot Simulation Components with TRL and IRL Assignments



TRL	Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	76	77	78	79	80	81	82	83
7	Physics Engine	1	7			1		7																				
7	Rendering Engine	2		7				7																				
4	Network Simulator Engine	3			7	2																			2			
2	Noise/Interference Models	4				7					3																	
7	Simulation Computing Soln	5	7	7	7		7	7																				
7	Simulation World	6	7	7				7	1	2	7	7	7	7				1	1									
2	SAR Perssonel Model	7							7			2		2														
2	Victim Models	8								7		2		2														
4	Terrain, Building Models	9									7	7	7	7														
7	Manually Generated Models	10										7			1													
4	Syntheticly Generated Models	11										1	7	1	1													
7	Model Database	12										7	1	7		7					2	2	2					
7	Realworld Data Collection	13													7													
7	Data Storage Devices	14												7	7	7												
7	Host Computing Soln	15															7			7								
2	Simulation Scenarios	16																7		1								
2	Simulation Logic	17																1	7									
7	Simulation GUI	18															7			7								
2	R-Spot Model	76										2		2							7							
2	R-UAV Model	77										2		2								7						
2	R- Ground Robot Model	78										2		2									7					
2	M-Coordination Algorithm	79																						7		2	2	
2	M-Mission GCS	80			2																			2	7	2	2	2
3	M-External Data Source	81																								7		
2	M-Human Mission Operator	82																							2		7	2
2	M-Authorities	83																							2		2	7

Figure 37. Multi-Robot Simulation Components with TRL and IRL Assignments