

Article

Unpacking the Role of Artificial Intelligence for a Multimodal Service System Design

Inkyung Sung ^{1,*}, Matthias Buderath ² and Peter Nielsen ¹

¹ Operations Research Group, Department of Materials and Production, Aalborg University, 9220 Aalborg, Denmark; peter@mp.aau.dk

² Airbus Defence and Space, 85521 Ottobrunn, Germany; matthias.buderath@airbus.com

* Correspondence: inkyung_sung@mp.aau.dk

Abstract: Since requirements of service demands are becoming increasingly complex and diversified, one of the success factors of a multimodal service system is its capability to design a specific service instance satisfying a specific set of requirements. This capability is further highlighted in Ad Hoc Multimodal Service Systems (AHMSSs), where service instances rarely follow a standard form of service delivery and exist only for a limited time. However, due to the increasing scale and frequency of services in many business and public sectors, meeting the desired level of capability has become troublesome. A well-designed Artificial Intelligence (AI) approach can be a solution to the difficulty by addressing the underlying complexity and uncertainty of the AHMSS design process. To conceptualize and foster AI applications to an AHMSS, this study identifies key decision-making problems in the AHMSS design process and discusses the role of AI in the process. The results will form the basis for AI development and implementation for an AHMSS and relevant service systems.

Keywords: artificial intelligence; machine learning; Ad Hoc Multimodal Service System; resource constraint project scheduling problem; decision-making automation; decision-making augmentation; decision support system



Citation: Sung, I.; Buderath, M.; Nielsen, P. Unpacking the Role of Artificial Intelligence for a Multimodal Service System Design. *Electronics* **2022**, *11*, 549. <https://doi.org/10.3390/electronics11040549>

Academic Editor: Dimitris Apostolou

Received: 13 January 2022

Accepted: 9 February 2022

Published: 11 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multimodal service systems deliver services to demands by using a combination of multiple means or resources of the systems. Due to these systems' flexibility in forming services with different configurations and capabilities, they have been actively implemented in various businesses and public sectors. A suitable example of this application is a multimodal transportation system, where a set of goods is transported via a combination of multiple means (container ships, trucks, trains, etc.) to meet demands with different geographical and operational conditions [1,2].

The flexibility in designing a service makes multimodal service systems more important in emergency and large-scale services, where how a service is delivered is not clearly defined until the demands occur and the desired services for the demands rarely follow a standard form of service delivery. Disaster relief [3], mass evacuation [4], and large-scale construction projects [5] are examples of these scenarios.

An Ad Hoc Multimodal Service System (AHMSS) is often designed and deployed to manage the urgency and rareness of demands. An AHMSS is a service instance designed specifically for a demand with certain requirements [6]. An AHMSS only lasts for a limited time period and serves a demand that is unique and new in terms of how the service is delivered. Figure 1 shows an example use case of an AHMSS in an emergency response situation, where a design for the AHMSS is performed as a decision-supporting function for an emergency response commander based on a request for an AHMSS and the available resource information.

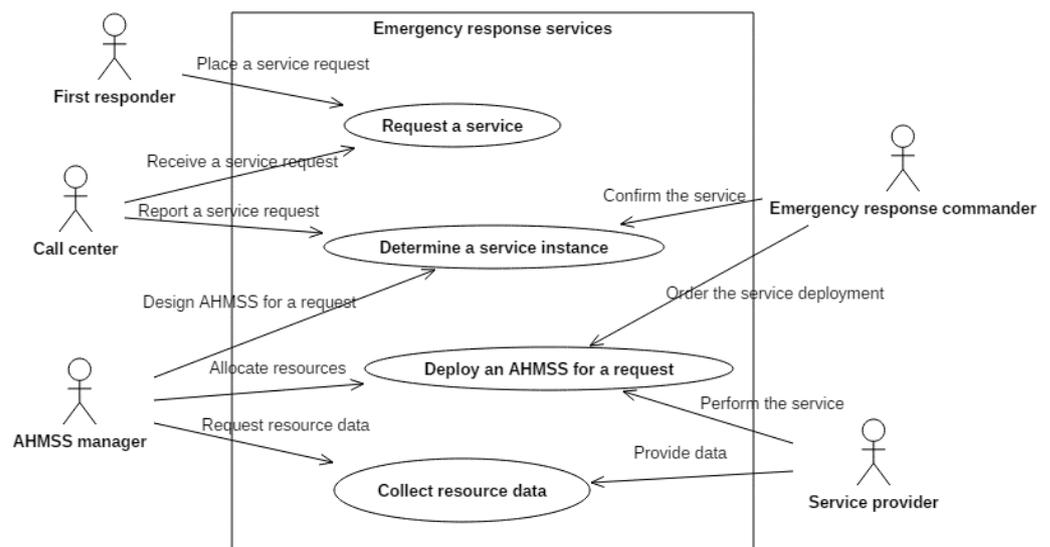


Figure 1. A use-case diagram for an AHMSS in an emergency response situation, reproduced from [6].

The design of an AHMSS requires multiple activities throughout the design process. First, requirements of a demand and available capabilities in a service system should be known or estimated to initiate the process with necessary data. After the initiation, an AHMSS configuration (a network of resources delivering a service) should be determined to fulfill the requirements of the demand by the capabilities of the configuration. Lastly, specific resources should be identified for the determined configuration so that the AHMSS can be deployed. In urgent and uncertain situations, where deployment conditions are continuously updated based on newly available observations, this process will be repeated until the service is completed.

Note that the AHMSS design process is challenging due to the nature of the situations where an AHMSS is needed. The main source of the challenge is the complexity of the decision-making problems involved in the process. One of the key decision-making problems in the AHMSS design process is to assign resources to a service considering their dependencies in forming a service capability. This problem can be formulated as a Resource Constrained Project Scheduling Problem (RCPSP), a strong NP-hard optimization problem [7]. Demand-specific requirements and the resulting constraints of the problem make the RCPSP even harder to solve. Having the RCPSP as a core decision-making problem, the AHMSS design process should also address other decision-making problems. For example, as an upstream problem of the RCPSP, service conditions and the available resource capabilities should be properly estimated so that an RCPSP instance can be generated in a manner reflecting the AHMSS deployment conditions.

Artificial Intelligence (AI) seems to be a good candidate for a solution to the challenge. Considering the nature of AHMSS, which demands immediate responses to a request, it is extremely difficult for human decision-makers to properly manage all the problems involved in the AHMSS design process without support. AI has demonstrated promising performances for many decision-making and -supporting problems. In particular, there have been solid advances in heuristics to solve complex optimization and decision-making problems [8]. Machine learning (ML) techniques have also shown their performances in many fields including classification, regression and even optimization [9].

Following the performance and potentials of AI, this study, as an extended work of Sung et al. [6], examines how AI can support the AHMSS design process. Sung et al. [6] discuss the necessity of designing a proper decision support system for AHMSS deployment. Based on this, we aim to further conceptualize the decision support system for AHMSS and to leverage the use of AI as the key solution for the challenges. To achieve this goal, we design a five-step AHMSS design process. This will serve as a framework into which various AI/ML techniques and their roles can be mapped for implementation.

Note that existing AI/ML studies in literature focus on specific parts of a decision-making process (e.g., optimization problem solving by AI and situational awareness creation by ML), neglecting a systematic and holistic perspective on the process. To avoid such a fragmented application of AI/ML to a decision-making process, the outcome of the systematic approach for the AHMSS design will be transformed into a contextual solution architecture that describes the AI functions and relevant system elements required to perform the functions for a decision-making process. This study is finally concluded with the expected challenges in the relevant AI development and implementation for an AHMSS.

2. Related Work

The majority of work in multimodal service systems has focused on steady-state systems or the systems with well-defined service delivery structures. Pawlewski [10] studies issues in applying multimodal transportation systems to supply chains where a standard network for the multimodal transport systems with fixed resources is defined. Bocewicz et al. [11] address a multimodal transportation system in a flexible manufacturing system with a given network for automated, guided vehicles. Bocewicz et al. [12] study a cyclic scheduling problem in a network with cyclic processes (e.g., subway and train traffic) to find the shortest path between two nodes.

Decision-making problems for multimodal service system design and planning have also received much attention. Hao and Yue [13] propose a dynamic programming-based optimization model to determine transport modes in a container multimodal transport system. Sitek and Wikarek [14] design an integer programming and constraint programming hybrid optimization approach for a supply chain with multimodal transport. Flórez et al. [15] propose a multimodal transportation planning system called *TIMEPLAN*, which derives (un)loading plans for goods and re-plans the original plans when trucks are damaged, new services enter, or a truck is delayed due to a traffic jam. Optimization approaches have been applied to recommend tours with multiple transportation modes and other considerations for tours (e.g., potential places to visit, cost, and traffic conditions) [16]. Please refer to SteadieSeifi et al. [2] for a detailed review on multimodal transportation and its strategic, tactical, and operational planning problems.

From the review on multimodal service systems, it can be concluded that the application of AHMSS and its design process have been rarely studied, despite increasing needs for AHMSS deployment. While there has been a lot of research on disaster relief logistics using multimodal service systems [3], where an AHMSS is suitable for application, the current state is rather focused on optimization problems with well-defined networks (location-allocation and routing problems) during a disaster response phase. These approaches require that the infrastructure is well-defined and that the resources, demands, and timing can be predefined.

The applications of AI to multimodal service systems are also limited to solving specific optimization problems (e.g., routing and scheduling) from a deterministic perspective. However, multimodal service system design and implementation involves solving a number of strongly interacting decision-making problems. Therefore, the entire process of the system design and planning should be addressed with a systematic view, which includes input data collection, problem instantiation (an instance creation for a design problem), problem solving, and decision making based on the output of the solution approach. Furthermore, considering the dynamics in the AHMSS-required situations (e.g., disaster response), uncertainties in service conditions should be addressed to some extent when designing an AHMSS so as to derive reliable and high-quality solutions. From the facts, it is concluded that the use of AI to design an AHMSS should be extended to support the AHMSS design process in a more systematic and comprehensive manner as well as to address the uncertainty and complexity issues in the AHMSS design process.

Based on the reviews and findings, the contributions of this study can be noted as follows. First, we conceptualize the entire process for an AHMSS design where key decision-making problems and relevant activities to solve the problems are identified. As a result,

difficulties in the decision making, especially for human decision-makers, are revealed, highlighting the needs for AI application to the AHMSS design process. AI-based solution algorithms and systems for the process are also discussed to illuminate the roles of AI in the process and foster the AI application to AHMSS and other relevant systems. These steps are fundamental and needed to enable the development of actual decision support systems.

3. Decision-Making Problems for an AHMSS Design

3.1. A Domain Model for an AHMSS

To fully understand the decision-making problems in an AHMSS design, we first present a domain model that visualizes the conceptual relationships between objects in an AHMSS, so the complexity of the AHMSS design can be illuminated. With the definitions of the main objects within an AHMSS, as shown in Table 1, the domain model for an AHMSS is presented in Figure 2.

Table 1. Main objects of an AHMSS, reproduced from [6]

Object	Definition
Service	A series of tasks requested by a customer
Customer	A person, group, or other entity who demands a service
Requirement	A condition for a service delivery
Resource	A system element that can contribute to service delivery
Configuration	A set of resources deployable together for a service
Configuration rule	A condition to form a configuration
Capability	An ability of a resource or a configuration to fulfill requirements
Availability	A state of a resource to be allocated for a service

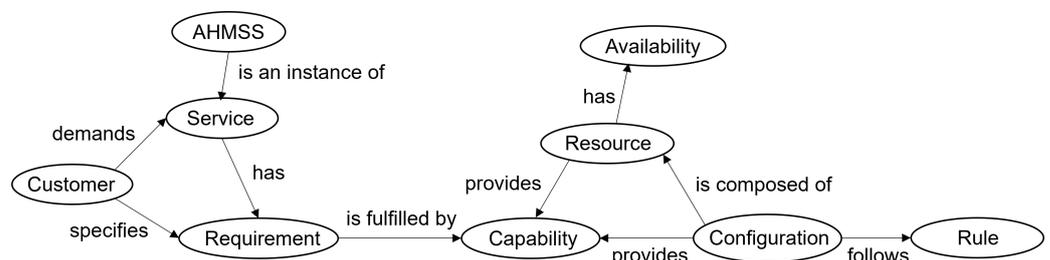


Figure 2. A domain model for an AHMSS, reproduced from [6].

The logical flow of the domain model is as follows. A customer demands a service, which has specific requirements (e.g., timing and location for a service). The requirements are fulfilled by the capabilities of a configuration (a set of resources) deployed for the service. A configuration can offer new or more capabilities than the simple sum of the individual resources’ capabilities. Imagine a drone with a camera, which can provide a better visualization with an aerial view. To form a configuration, a certain set of rules should be respected. A precedence relationship and compatibility between resources are examples of the rules. Resources are further represented as their availability over time.

3.2. An AHMSS Configuration Problem

From the domain model, an AHMSS configuration problem can be identified, which determines a configuration following its configuration rules so that the capabilities of the configuration can fulfill the requirements of a service demand. More specifically, each service may task different operations, resulting in different requirements in set $Q = \{q_1, \dots, q_p\}$. Each resource in set $R = \{r_1, \dots, r_m\}$ may also have different capabilities in set $C = \{c_1, \dots, c_k\}$, which are affected by the conditions of an operational environment in set $E = \{e_1, \dots, e_u\}$.

Given the setting that requirements are only met by the capabilities of a configuration, the AHMSS configuration aims to determine configuration G as a sub-set of R ($G \subseteq R$),

which is transformed to a set of capabilities $C' \subseteq C$ by function $\mathcal{F} : 2^m \mapsto 2^k$. Function \mathcal{F} takes a subset of E as input to map configuration G to capability set C' based on operational conditions.

Requirement q may require a set of capabilities $C' \subseteq C$ to be met. Thus, the set of requirements for service s , $Q_s \subseteq Q$, can be transformed to a set of sets of capabilities, $\{C'_1, \dots, C'_v\}$, where $C'_i \subseteq C$, $1 \leq i \leq v$, by function $\mathcal{G} : 2^q \mapsto 2^{2^k}$.

Lastly, given a set of alternative configurations producing different sets of capabilities, we can define decision-making function \mathcal{M} , which determines the optimal configuration G^* such that the required capabilities of service s can be provided by the optimal configuration, i.e., $\mathcal{G}(Q_s) \subseteq \mathcal{F}(G^*)$. The transitions between domain objects by the functions described are illustrated in Figure 3.

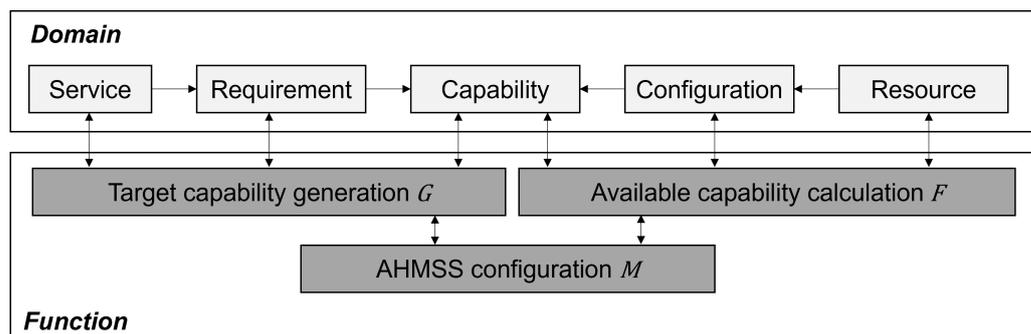


Figure 3. A domain object transitions for an AHMSS configuration, reproduced from [6].

3.3. A Resource Constrained Project Scheduling Problem

Note that the AHMSS configuration problem addresses how the capability of a configuration is formed without considering the availability of the resources assigned to the configuration. To deploy a configuration, the resources of the configuration should be available during the operation time windows for a service. An operation time window of a resource is a time interval between the start and end of operations performed by the resource in an AHMSS.

An operation time window of a resource is determined by the role of the resource within a configuration and the dependency with other resources of the configuration. For example, a container crane and a berth at a container terminal should be available at the same time and for a whole period of container (un)loading to handle a container ship. As such, resource allocation timing should be determined along with a configuration decision so that the deployment of the determined configuration can be feasible.

From optimization perspectives, the decision-making problem and constraints involved in the AHMSS design can be formulated as an RCPSP. In the RCPSP, a project, which corresponds to a service in the AHMSS context, consists of multiple tasks, where a task is an activity to fulfill requirements for a service. A task has a precedence relationship with other tasks of the project, which represents how a service is delivered. To complete a task, capable resources of the task should be assigned. With the setting, the RCPSP is often solved to minimize the completion time of a project while allocating available resources to the project subject to resource availability. A solution to the RCPSP (a resource allocation schedule) eventually forms a configuration for a service.

Given that the RCPSP belongs to a class of NP-hard problems, a decomposition approach that divides the problem into two stage problems (a master problem for a resource assignment and a sub-problem for resource scheduling) would be appropriate. This would allow one to manage the complexity of the RCPSP and guarantee the quality of solutions finally derived [17]. Detailed descriptions and formulations for this decomposition can be found in Appendix A.

4. AI for the Entire AHMSS Design Process

4.1. Niches for AI Application: The Complexity and Uncertainty in the AHMSS Design

As noted, the decision-making problems in the AHMSS design are complex. The complexity will be further complicated when addressing supporting resources that make the main service resources able to function. The dependencies between resources and their capabilities will result in complicated temporal constraints on tasks and resource-related variants of the RCPSP. Figure 4 shows examples of service enablers necessary for a container transportation with two means (a container ship and a truck) and their relationships.

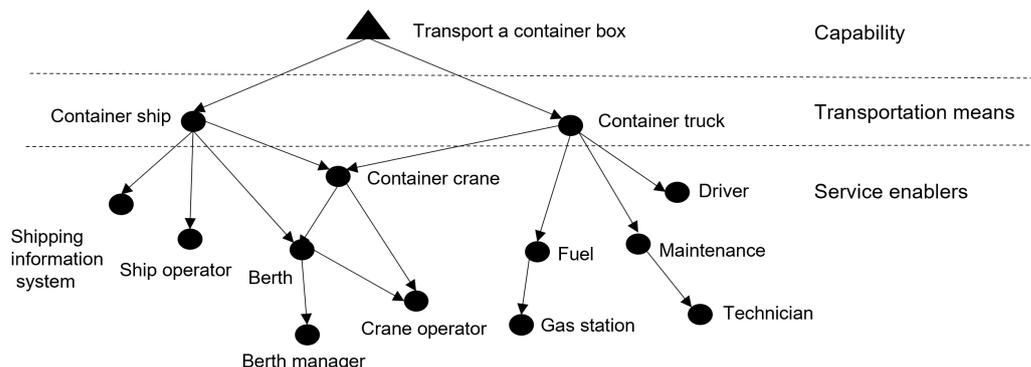


Figure 4. Example dependencies between service elements for container transportation, reproduced from [6].

The uncertainty in a situation where an AHMSS is deployed is also significant due to a lack of information about operational environments. In particular, upon a request for an AHMSS, a creation of a clear understanding about a target request would be almost impossible. In the aftermath of fatal flooding in western Germany in July 2021, for example, communication was broken, making it difficult to properly estimate the magnitude of the disaster and the corresponding scale of response efforts [18].

It should be clearly noted that the complexity and uncertainty in decision making are the key niches for AI-based decision-making/supporting systems and their application [19]. More specifically, the complexity of a decision-making process can be fully addressed by AI with its great computational and analytical capacities. Regarding the uncertainty in decision making, AI’s capability to provide real-time information access and predictive analytics can support humans’ cognition and intuitive decisions.

4.2. The AHMSS Design Process

To clearly understand the AHMSS design process and provide a pragmatic perspective on the role of AI to the process, we first introduce an entire AHMSS design process, as illustrated in Figure 5.

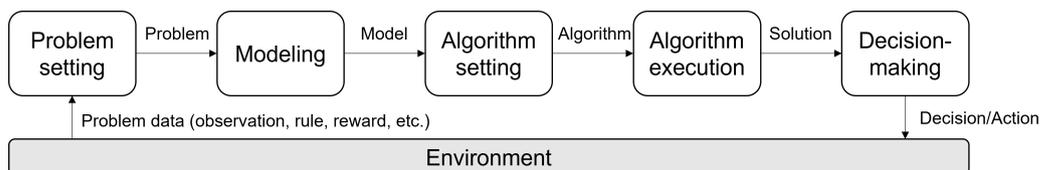


Figure 5. The AHMSS design process.

4.2.1. Step 1. Problem Setting

The first step of the AHMSS design process is the problem setting that identifies a target AHMSS design problem given problem data. The data include observations on an operational environment and rules for resource allocation. If the process is re-initiated by an action performed based on solutions from the previous process, the observed rewards or updated situational awareness about the environment are included in the problem data.

In the step, a goal and how the goal achievement is measured are first defined and relevant objective functions, decisions to make, and constraints that control the decisions are defined based on the goal, resulting in an AHMSS design problem.

4.2.2. Step 2. Modeling

The problem defined in step 1 is then abstracted to a model, a mathematical or logic object. Linear programs and constraint programs are examples of such models. Problem contexts should be considered in this modeling step so that important features of the problem can be incorporated into the modeling and the resulting solutions. The problem-solving capacity and how the solution can be translated into a decision are also important factors to consider in this step to obtain acceptable solutions within a reasonable computation time. In general, incorporating as many features of a problem as possible into a model increases the complexity of the modeling and the resulting solution algorithm, whereas neglecting the features during modeling will reduce the applicability of the final solution.

4.2.3. Step 3. Algorithm Setting

The model formulated is solved by a solution algorithm or system. A solution algorithm is designed to explore a solution space of the model (a set of feasible solutions to the model) and evaluate the solutions found with respect to the objective function of the model. For a constraint program that focuses on finding feasible solutions, the solution evaluation can be omitted in the solution algorithm. How a solution algorithm explores a solution space is often controlled by parameters of the algorithm. A solution representation (e.g., a bit-string for binary decision variables) and initialization of the algorithm (where to start the solution search) are also the variables of the algorithm setting.

4.2.4. Step 4. Algorithm Execution

This step runs the algorithm to find a solution to a particular problem instance. The algorithm can be run with different experimental settings, resulting in multiple algorithm running instances. Parallel algorithm execution can be considered when the computation time to solve the problem is limited. Specific termination conditions for the algorithm run (e.g., the maximum computation time budget) can also be set based on the context of the problem in this step.

4.2.5. Step 5. Decision-Making

The solution(s) found by the algorithm execution is translated into a decision. In the context of the AHMSS design, the decision is about which configuration and which resources at what time are deployed. Given that the intuition and experiences of a human in-the-loop are determinants of the decision making in many systems, the solutions and their performances evaluated at step 4 can serve as cognitive assistance and decision-supporting functions.

4.3. Using AI for the AHMSS Design

The role of AI for decision making can be categorized into two classes: *automation* and *augmentation*. AI-based automation of decision making is often implemented for decision-making problems with complicated computations with well-defined domain knowledge, whereas AI-based augmentation of decision making is dominantly implemented when decision-making problems involve relatively high uncertainties and ambiguities [20]. Following the classification, Table 2 summarizes the key use of AI in the AHMSS design process with respect to the two classes.

For the problem setting step, AI can automate data-related tasks. The available data for the problem setting are likely large-scale due to a large volume of sensor and monitoring data. Therefore, it is difficult for humans to handle the data properly in limited time. The tasks include handling missing data using approximation or interpolation of the available dataset and integrating multiple data sources into unified and useful datasets. With the

dataset prepared, AI can be further used for cognitive supports. Detecting patterns in data and interpreting the patterns are examples of the AI-based cognitive supports.

Table 2. A summary of AI usage in the AHMSS design.

Step	Automation	Augmentation
Problem setting	Data preparation Data fusion	Pattern detection Predictive analytics Interpretation
Modeling	Modeling automation	Scenario generation Model selection
Algorithm setting	Optimization Parameter tuning Algorithm selection	-
Algorithm execution	Computing capacity allocation Simulation	Simulation (virtual representation)
Decision-making	Knowledge transfer	Solution explanation Solution comparison

In the modeling step, AI can help humans decide how to model the problem identified. Modeling features (e.g., deterministic vs. stochastic) and the generation of scenarios that a solution will be evaluated under are example decision elements. AI can also automate the task to create a model that a machine can read and work on using natural language processing and a unified mathematical and logical interface.

The use of AI in the algorithm setting step is biased to the automation class by nature of the step. For combined optimization and constraint satisfaction problems, the state-of-the-art heuristics and meta-heuristics have demonstrated the capability to solve the problems in a reasonable computation time with near-optimality for even relatively large problem instances. Meta-optimization that uses an algorithm to tune another optimization algorithm can be implemented to customize a solution algorithm per problem instance so that the solution algorithm can be run in its best-working condition. ML algorithms are often used for this purpose. Please see Karimi-Mamaghan et al. [21] for a detailed review of the use of ML for meta-heuristics.

In the algorithm execution step, AI can decide how to allocate the available computing power to the multiple algorithm instances tuned with different configurations. A simulation can be embedded into a solution algorithm to evaluate a solution under uncertainties (modeled as scenarios or distributions). Simulation techniques are also key to virtually represent an operational environment and the effects of a solution so that humans can understand what would happen with the AI solutions.

Finally, the obtained solutions can be interpreted using AI for decision making. This aspect is essential for an AHMSS design where humans in-the-loop may handle life-threatening decisions and are to understand and trust the inputs from AI. Following the importance, explainable AI (XAI) has received much attention [22]. Interpretable models, such as random forests, or techniques that infer an explainable model on top of a black box model can be used to develop XAI [23,24]. Furthermore, the outcomes of AI over all the steps of the AHMSS process can be transferred to next processes to fully exploit the findings and lessons from the current stage in the following processes. By doing so, the quality of the AI outcomes can be improved while saving resources for AI creation and training. This knowledge transfer is critical when there is a lack of data and time to train ML algorithms.

Based on the use of AI for the AHMSS design, a disaster response scenario with the AI-based decision support system is illustrated in Figure 6. In the figure, an AI-based decision support system continuously interacts with humans in-the-loop, providing relevant information for decision making throughout the AHMSS design process.

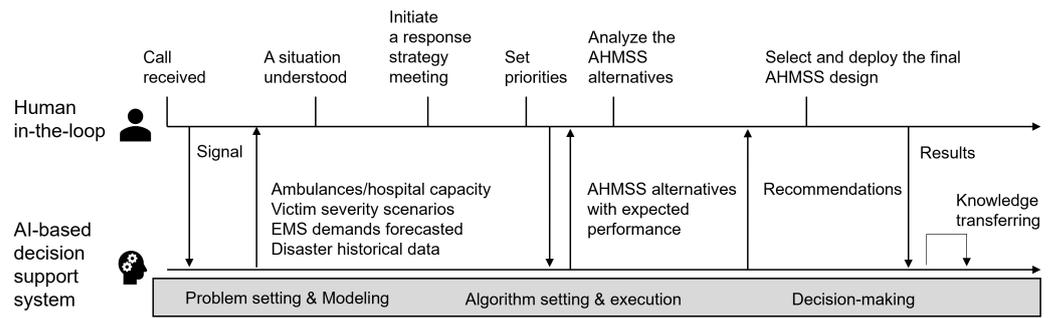


Figure 6. An example disaster response scenario with an AI-based decision support system.

5. A Solution Architecture for AI Functionality Implementation

Understanding behaviors of AI functions for a decision-making process and identifying the relevant system elements required to perform the functions properly are a key step in developing an AI-based decision support system. To support the key step, upon understanding the use of AI in the AHMSS design process as a basis, we describe an AI functional block according to the four dimensions of a functional description—input, output, control, and resource—as proposed in the Integration Definition for Function Modeling (IDEF0) approach [25].

The input for an AI function is the information or data required to formulate a problem for the function. For learning-based AI functions, the input would also be used to train the algorithm implemented. Here, an AI-based decision support system requires a database or data interfaces with external data sources to feed the AI function with relevant inputs. Controls of the AI function are rules and policies that govern the function. Controls are set by human decision-maker’s objectives or preferences; therefore, an interface to interact with humans in-the-loop is important to guide the AI function as desired. Given the input and control of the AI function, computing resources (including hardware and software) will be assigned to perform the function. The output of the function is the information or data generated by the function. To support human’s decision making, the output will be forwarded to the human–computer interface in a human readable format. The output will also be forwarded to the database for the following AI functions.

Based on the AI function description, a contextual solution architecture for an AI-based decision support system with the relevant system elements can be drawn as in Figure 7.

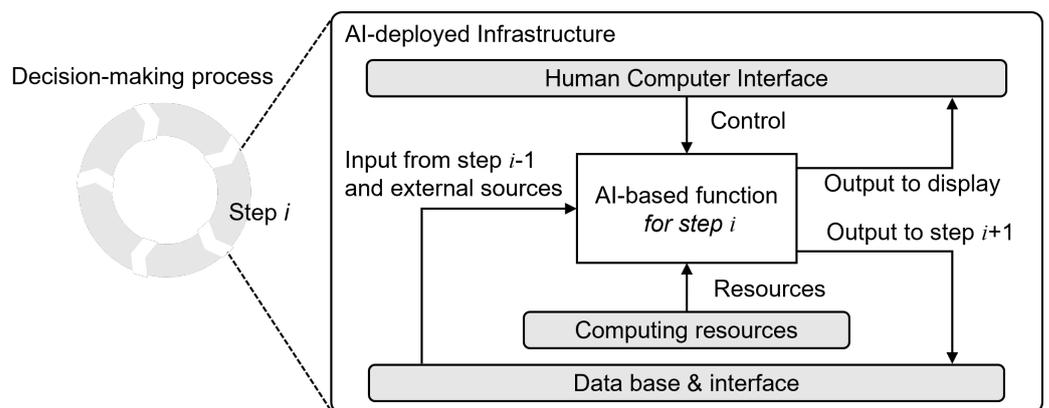


Figure 7. A solution architecture design with AI-based functions.

6. Challenges in AI Application to the AHMSS Design

While the advances in AI and relevant technologies have demonstrated the capability for replacing or supporting human decision-makers as intelligence partners, the development and implementation of AI for the AHMSS design process with desired functionalities, as described in Section 4, are still challenging due to the issues in integrating an AI-based decision support system into a human-dominated legacy system.

First, the requirements of a service and constraints/rules on the service delivery may not be stated in a structured and machine-readable format for ad hoc service systems. They are straightforward to describe in controlled environments, such as manufacturing and ordinal transportation systems. However, such a structure can seldom be defined for ad hoc systems. This fact prohibits an AI-based system from automating some of its featured functionalities and thus fast responses to service requests.

Next, humans are not willing to work with machines unless humans control the majority of decision making. According to Haesevoets et al. [26], most managers will accept AI decisions when they feel the decision is made on primarily their input and judgment. There are also people who strongly oppose the AI implementation, especially for a managerial decision.

Here, XAI plays a key role. However, many attempts for XAI development have suffered from performance degradation. The majority of ML techniques that lead recent AI success, such as deep learning algorithms, involve internal layers, are hard to understand, and are interpreted from humans' perspective. On the other hand, most explainable algorithms, such as decision trees, often have the least performance [23]. Therefore, AI/ML techniques with acceptable performance and explainability should be studied and developed to foster human–AI combined decision-making environments.

Humans' barriers to accept machines' decisions also highlights the way to put humans and machines together for decision making, exploiting their own strength. Characterizing the strength of the players into three classes—uncertainty, complexity, and equivocality—Jarrahi [19] proposes that AI can be positioned in a decision-making process to tackle complexity issues using its computing power, whereas humans can contribute more to handling uncertainty and equivocality in the process using creative and intuitive approaches.

Lastly, most AI and ML techniques require a software deployment environment with high levels of computing power and data management capability. Moreover, in the AHMSS design process, information about resources available in the system would be owned by multiple stakeholders, especially when the AHMSS covers various capabilities and acts as a centralized control function for multiple sub-service systems. Given that a large volume of training data is needed for learning-based algorithms, a cloud-based data storage system can be considered for the software environment of the AHMSS design process. A cloud computing service is also suitable for the environment since the computational burden of running the optimization intelligence involved is expected to be high and the speed of the AHMSS design process should be fast.

7. Summary

AI has been the focus of solution systems for various decision-making problems following AI's success in many fields. The same is applied to the AHMSS design process where the decision-making capability for service configuration and resource allocation in limited time is essential. A number of resources and complicated relationships between the resources in forming a service capability further underline the needs for AI in order for the process to handle the complexity of the involved decision-making problems. The characteristics of a situation where an AHMSS is suitable also highlight the use of AI to manage the uncertainty in the situation. Given that humans excel in handling uncertainties and ambiguities of a situation using their intuition, experiences, and ability to learn from unrelated situations, AI's capability in detecting underlying and hidden patterns in service delivery can augment humans' decision making.

To promote AI application in the AHMSS design process, we identified key decision-making problems in the process, the AHMSS configuration, and the RCPSP. Current AI techniques, especially for optimization, have shown their performance in solving such problems with (near) optimality within a reasonable computation time. It should be noted that the use of AI is not limited to solving the decision-making problem. Upon understanding the AHMSS design process, which is divided into five steps: problem setting, modeling, algorithm setting, algorithm execution, and decision making, the feasible

and promising roles of AI for these five steps are discussed with respect to automation and augmentation of decision-making.

We believe the proposed conceptual and contextual work contributes to identifying a gap in the current use of AI in a decision-making process and forming a basis for an AI-based decision support system. To further support and prove the contributions of the work, future work will focus on implementing an AI-based decision support system for a specific use case following the proposed solution architecture.

Author Contributions: Conceptualization, I.S., M.B. and P.N.; Methodology, I.S., M.B. and P.N.; Formal analysis, I.S.; Visualization, I.S.; Writing—original draft, I.S.; Writing—review and editing, I.S. and P.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially funded by Airbus Defence & Space.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. A Decomposition Approach for the RCPSP

The master problem of the RCPSP is designed to assign resources to tasks considering the resources' capacity (total available time period) and capability to perform the tasks. The decision made by solving the master problem is then delivered to the sub-problem, where the timing for the resource allocation is determined subject to the time-dependent resource availability and the precedence constraints between tasks. If the sub-problem cannot find a feasible solution with the decision from the master problem, the master problem is solved again with a cut that excludes the previous solution from its solution space.

The master problem can be formulated as an extended multiple knapsack problem that assigns a set of n tasks (items) to a set of m resources (knapsacks) to maximize the total number of tasks assigned to resources subject to the capacity of the resources and the resource compatibility for a configuration. The formulation of the master problem can be written as

$$\max \sum_{i=1}^n \sum_{j=1}^m x_{ij} \quad (\text{A1})$$

$$\text{subject to } \sum_{i=1}^n d_{ij}x_{ij} \leq c_j \quad \forall j \in \{1, \dots, m\}, \quad (\text{A2})$$

$$\sum_{j=1}^m x_{ij} \leq 1 \quad \forall i \in \{1, \dots, n\}, \quad (\text{A3})$$

$$x_{ij} + x_{i'j'} \leq 1 \quad \forall (i, i', j, j') \in C, \quad (\text{A4})$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in \{1, \dots, n\}, j \in \{1, \dots, m\}. \quad (\text{A5})$$

The decision variable x_{ij} equals one if task i is performed by resource j ; otherwise, it is zero. With this decision variable, the objective function (A1) is set to maximize the number of tasks assigned to available resources. A weight can be used to incorporate the different importance and priorities of tasks for a service, which changes the objective function to maximize a weighted sum of resource allocation decisions.

d_{ij} is the duration of task i by resource j , and c_j is the capacity of resource j (a total length of available periods). With the parameters, Constraint (A2) ensures that the total duration assigned to a resource is less than or equal to the capacity of the resource. If resource j is not capable of performing task i , d_{ij} is set as a large number to avoid the corresponding task allocation decision. Constraint (A3) allows a task to be allocated to a resource at most once. C is the set with the pairs of task-resource allocation decisions,

which cannot be performed together. With the set, Disjunctive Constraint (A4) ensures the compatibility constraint between resources. Constraint (A5) is a binary constraint for the decision variable x .

The task-resource allocation decision x is then delivered to the sub-problem that determines the starting and end time of the tasks of a service subject to the resource availability and the precedence relationship between tasks. The constraints of the sub-problem can be conceptually represented by

$$T_i - d_i \geq \tilde{T}_{ki} \quad \forall (k, i) \in E, \quad (\text{A6})$$

$$\sum_{i \in S_t} x_{ij} \leq a_{jt} \quad \forall t = 1, 2, \dots, T_n, j \in \{1, \dots, m\}. \quad (\text{A7})$$

Constraint (A6) is the temporal constraint between two tasks. T_i is the completion time of task i , d_i is the duration of task i , and \tilde{T}_{ki} is the reference starting time for task i in relation to task k . \tilde{T}_{ki} will be calculated based on the temporal relationship between tasks k and i , which includes the set-up time between the tasks, generalized precedence relationships (start-to-start, start-to-end, end-to-start, and end-to-end), and time windows for a task completion. If there is a simple end-to-start precedence relationship between tasks k and i with a zero time lag, the reference starting time \tilde{T}_{ki} can be determined by the completion time of task k , T_k . The time lags between two tasks' operation time may depend on the resource assignment to the tasks [27].

Constraint (A7) ensures that resources are assigned to tasks without schedule conflicts. In the constraint, S_t denotes the set of activities in progress in time interval $(t - 1, t]$, and a_{jt} is the availability of resource j at time t . With this setting, the total number of resources j in use at a certain given time should follow the availability of the resource at that time.

The sub-problem may focus on the feasibility of resource scheduling given a resource allocation decision by the master problem. That said, the makespan of the service and other objective functions can be implemented in the sub-problem to select a best-fit solution among alternative feasible resource schedules.

When the sub-problem fails to find a feasible solution, the master problem will be solved again, excluding the previous resource allocation solution. To implement this scheme, a logic-based Benders decomposition can be applied to generate a cut for the master problem [28]. With the concept, a simple logical expression, a *no-good* cut, is used to represent the feasibility and optimality of a trial solution. For a problem with a binary decision variable vector x , an example of a no-good cut that excludes an infeasible solution \tilde{x} can be given by

$$\sum_{i \in T} x_i - \sum_{i \in F} x_i \leq |T| - 1, \quad (\text{A8})$$

where $T = \{i | \tilde{x}_i = 1\}$ and $F = \{i | \tilde{x}_i = 0\}$ [29].

References

1. Cipriani, E.; Gori, S.; Petrelli, M. Transit network design: A procedure and an application to a large urban area. *Transp. Res. Part C Emerg. Technol.* **2012**, *20*, 3–14. [CrossRef]
2. SteadieSeifi, M.; Dellaert, N.P.; Nuijten, W.; Van Woensel, T.; Raoufi, R. Multimodal freight transportation planning: A literature review. *Eur. J. Oper. Res.* **2014**, *233*, 1–15. [CrossRef]
3. Tariq, S.; Jalil, M.N.; Zaffar, M.A. Multimodal logistics in disaster relief. In *The Palgrave Handbook of Humanitarian Logistics and Supply Chain Management*; Palgrave Macmillan: London, UK, 2018; pp. 445–487.
4. Naghawi, H.; Wolshon, B. Operation of multimodal transport system during mass evacuations. *Can. J. Civ. Eng.* **2015**, *42*, 81–88. [CrossRef]
5. Asadi, K.; Suresh, A.K.; Ender, A.; Gotad, S.; Maniyar, S.; Anand, S.; Noghabaei, M.; Han, K.; Lobaton, E.; Wu, T. An integrated UGV-UAV system for construction site data collection. *Autom. Constr.* **2020**, *112*, 103068. [CrossRef]
6. Sung, I.; Buderath, M.; Nielsen, P. A Structural and Functional Design for Resource Deployments in Ad Hoc Multimodal Service Systems. In Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence, Salamanca, Spain, 6–8 October 2021; pp. 31–40.
7. Hartmann, S.; Briskorn, D. A survey of variants and extensions of the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **2010**, *207*, 1–14. [CrossRef]

8. Ng, K.; Lee, C.K.; Chan, F.T.; Lv, Y. Review on meta-heuristics approaches for airside operation research. *Appl. Soft Comput.* **2018**, *66*, 104–133. [[CrossRef](#)]
9. Gambella, C.; Ghaddar, B.; Naoum-Sawaya, J. Optimization problems for machine learning: A survey. *Eur. J. Oper. Res.* **2021**, *290*, 807–828. [[CrossRef](#)]
10. Pawlewski, P. Multimodal approach to model and design supply chain. *IFAC Proc. Vol.* **2013**, *46*, 2110–2115. [[CrossRef](#)]
11. Bocewicz, G.; Banaszak, Z.; Nielsen, I. Multimodal processes prototyping subject to fuzzy operation time constraints. *IFAC-PapersOnLine* **2015**, *48*, 2103–2108. [[CrossRef](#)]
12. Bocewicz, G.; Nielsen, P.; Banaszak, Z.A.; Dang, V.Q. Cyclic steady state refinement: Multimodal processes perspective. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Stavanger, Norway, 26–28 September 2011; pp. 18–26.
13. Hao, C.; Yue, Y. Optimization on combination of transport routes and modes on dynamic programming for a container multimodal transport system. *Procedia Eng.* **2016**, *137*, 382–390. [[CrossRef](#)]
14. Sitek, P.; Wikarek, J. A hybrid approach to modeling and optimization for supply chain management with multimodal transport. In Proceedings of the 2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 26–29 August 2013; pp. 777–782.
15. Flórez, J.E.; de Reyna, A.T.A.; García, J.; López, C.L.; García-Olaya, Á.; Borrajo, D. Planning multi-modal transportation problems. In Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling, Freiburg, Germany, 11–16 June 2011.
16. Lim, K.H.; Chan, J.; Karunasekera, S.; Leckie, C. Tour recommendation and trip planning using location-based social media: A survey. *Knowl. Inf. Syst.* **2019**, *60*, 1247–1275. [[CrossRef](#)]
17. Lombardi, M.; Milano, M. Optimal methods for resource allocation and scheduling: A cross-disciplinary survey. *Constraints* **2012**, *17*, 51–85. [[CrossRef](#)]
18. Jenny, H. Europe floods: At least 120 dead and hundreds unaccounted for. *BBC*, 2021. Available online: <https://www.bbc.com/news/world-europe-57858829> (accessed on 10 January 2022).
19. Jarrahi, M.H. Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Bus. Horizons* **2018**, *61*, 577–586. [[CrossRef](#)]
20. Duan, Y.; Edwards, J.S.; Dwivedi, Y.K. Artificial intelligence for decision making in the era of Big Data—evolution, challenges and research agenda. *Int. J. Inf. Manag.* **2019**, *48*, 63–71. [[CrossRef](#)]
21. Karimi-Mamaghan, M.; Mohammadi, M.; Meyer, P.; Karimi-Mamaghan, A.M.; Talbi, E.G. Machine Learning at the service of Meta-heuristics for solving Combinatorial Optimization Problems: A state-of-the-art. *Eur. J. Oper. Res.* **2022**, *296*, 393–422. [[CrossRef](#)]
22. Chou, Y.L.; Moreira, C.; Bruza, P.; Ouyang, C.; Jorge, J. Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. *Inf. Fusion* **2022**, *81*, 59–83. [[CrossRef](#)]
23. Gunning, D.; Aha, D. DARPA’s explainable artificial intelligence (XAI) program. *AI Mag.* **2019**, *40*, 44–58.
24. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [[CrossRef](#)]
25. Lightsey, B. *Systems Engineering Fundamentals*; Technical report; Defense Acquisition University: Fort Belvoir, Virginia, 2001.
26. Haesevoets, T.; De Cremer, D.; Dierckx, K.; Van Hiel, A. Human-machine collaboration in managerial decision making. *Comput. Hum. Behav.* **2021**, *119*, 106730. [[CrossRef](#)]
27. Sabzehparvar, M.; Seyed-Hosseini, S.M. A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags. *J. Supercomput.* **2008**, *44*, 257–273. [[CrossRef](#)]
28. Hooker, J.N.; Yan, H. Logic circuit verification by Benders decomposition. In *Principles and Practice of Constraint Programming: The Newport Papers*; MIT Press: Cambridge, MA, USA, 1995; pp. 267–288.
29. Jain, V.; Grossmann, I.E. Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS J. Comput.* **2001**, *13*, 258–276. [[CrossRef](#)]