Self-Adaptation in Human-in-the-Loop Cyber-Physical Systems Using Digital Twins

By

Hargyo Tri Nugroho Ignatius

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham

June 2024

Abstract

Human-in-the-Loop Cyber-Physical Systems (HitLCPS) allow individuals to assume diverse roles within the system, interacting with various Cyber-Physical System (CPS) components to achieve common objectives. Engineering HitLCPS presents several challenges. Firstly, orchestrating services involving humans and machines as service providers requires considering human values and the distinct characteristics of humans and machines. Secondly, as self-adaptive systems, CPS must adapt to environmental contexts characterised by various uncertainties to achieve their goals.

This thesis presents a conceptual reference model for HitLCPS that considers the diverse roles of humans in the CPS and their characteristics, which involve human values. We also demonstrate how to incorporate three distributive justice principles into task allocation in human-machine contexts. Lastly, we propose a framework for decision-making under uncertainty resulting from different levels of observability in the environment. Specifically, we introduce a multi-reward Markov decision process for mixed observabilities, which is evaluated using scenarios such as remote data mirroring and credit card payment systems.

Digital twin architecture offers flexibility for experimenting with various what-if scenarios and simulating scenarios that real-world datasets may not capture. We utilise digital twins to conduct comprehensive evaluations and to enable self-adaptation in HitLCPS through our proposed methods.

Evaluation results indicate that our proposals perform favourably compared to the baselines. Our reference model provides a high level of completeness, adaptability, accuracy, clarity, and consistency; our multi-objective fairness approach yields better overall fairness in various scenarios; and our decision-making framework can better satisfy the non-functional requirements and provide better trade-offs.

Ad majorem Dei gloriam

To my parents and family, whose endless love and encouragement have been my greatest blessings.

Acknowledgements

Praise be to the Almighty Father for giving me strength, sending good people around me, and allowing me to finish this thesis.

I extend my thanks to the government of the Republic of Indonesia through the Indonesia Endowment Fund for Education (LPDP) and Universitas Multimedia Nusantara for funding my PhD.

I am extremely grateful to my supervisor, Dr Rami Bahsoon, for his exceptional guidance, wisdom, and unwavering support throughout this journey. In addition to being my research and academic supervisor, Rami has backed me through the challenging times of my PhD path as a friend and mentor.

I am indebted to my thesis committee members, Dr Leandro Minku and Dr Shan He, for their insightful discussions, valuable comments, and constructive feedback during our RSMG meetings.

I would also like to express my gratitude to my collaborators, Dr Huma Samin and Dr Nelly Bencomo, for introducing me to MR-POMDP and sharing the code generously and also for their valuable feedback in developing SPECTRA, which has been an essential part of this thesis. Knowing and collaborating with both of them has been an honour and privilege.

I am deeply grateful to my colleagues in the Software Engineering Research Group: Hayatullahi Adeyemo, Hanouf Alghanmi, Dr Nan Zhang, Dr Akram Alofi, Dr Sabreen Ahmadjee, Dr Wendy Yanez, Dr Satish Kumar; my office-mates in room 144: Nawfal Zakar, Bakhyt Bakiyev, Dr Shatha Hakami, Dr Fatma Faruq, etc.; as well as my friends from Indonesia whom I cannot mention one by one. Thank you for the wonderful times, meaningful discussions, and laughters.

Last, I express my deepest gratitude to my beloved family, whose unwavering love, support, and encouragement have guided me throughout this journey. To my wonderful wife, Jane, and children, thank you for your patience, understanding, and sacrifices. Your constant presence and belief in me have strengthened and motivated

me. To my awesome mother and mother-in-law, my sisters and brothers-in-law, and the big family of Poedjojoewono, thank you for your love, prayers, and support.

Contents

List of Figures x

List of Tables xv

1 Introduction 1

1.1 Motivation . . . 1

1.2 Problem Statement . . . 3

1.3 Research Questions . . . 5

1.4 Thesis Objectives and Contributions . . . 5

1.5 Publications . . . 7

1.6 Thesis Roadmap and Storyline . . . 8

2 Self-Adaptation and Digital Twins for HitLCPS 13

2.1 Self-Adaptive Cyber-Physical Systems . . . 14

2.1.1 Definitions . . . 14

2.1.2 Adaptation Properties . . . 15

2.1.3 Adaptation Pattern . . . 17

2.2 Human-in-the-Loop CPS . . . 21

2.2.1 Human vs Machine . . . 22

2.2.2 Taxonomy of HitLCPS . . . 24

2.2.3 Human roles in HitLCPS . . . 25

2.2.4 Distinguishing Characteristics and Challenges of HitLCPS . . 27

2.3 Digital Twin for Adaptation in HitLCPS . . . 28

2.3.1 Digital Twins vs. MAPE-K 29
2.3.2 Deployment 31
2.3.3 Temporal Integration 32
2.3.4 Twinning Type 32
2.3.5 Feedback Type 33
2.3.6 Digital Twin Levels 33
2.3.7 Layers and Human Modes 35
2.4 Identified Gaps in HitLCPS 36
2.4.1 Cognition/Capability 37
2.4.2 Motivation 40
2.4.3 Predictability 42
2.5 Summary 45

3 A Conceptual Reference Model for HitLCPS **47**

3.1 Introduction 48
3.2 *-as-a-Service in HitLCPS 50
3.2.1 Everything as a Service 50
3.2.2 Human-as-a-Service 51
3.3 The O*NET Framework 53
3.4 Proposed SOA-HitLCPS Ontology Model 55
3.4.1 Service Model 56
3.4.2 Service Profile 57
3.4.3 Human Capability 58
3.4.4 Machine Capability 61
3.5 Using our Ontology in Self-adaptive HitLCPS 61
3.5.1 Architecture 62
3.5.2 Example Scenarios 63
3.6 Evaluation 69
3.7 Threats to Validity 73

3.8 Conclusion 74

4 Fairness-Aware Service Provisioning in HitLCPS: a Heterogenous Crowdsourcing Case 75

4.1 Introduction 76
4.2 Fairness and Distributive Justice Principles 78
4.2.1 Equity 79
4.2.2 Equality 80
4.2.3 Need 80
4.3 Motivating Scenario 81
4.4 Problem Definition 84
4.4.1 Fairness requirements 84
4.4.2 Preliminaries 85
4.4.3 Unfairness Indices 88
4.4.4 Problem formulation 90
4.5 Fairness-Aware Online Task Allocation 90
4.6 Digital Twin Architecture of Heterogenous Crowdsourced Logistics . 93
4.6.1 Digital Twin Layer 94
4.6.2 Asset Layer 95
4.6.3 Decision Layer 96
4.7 Evaluation and Discussion 96
4.7.1 Experiment Setup 97
4.7.2 Comparative Approaches 98
4.7.3 Digital Twin at Design Time 100
4.7.4 Digital Twin for Runtime Adaptation 104
4.8 Threats to Validity 109
4.9 Conclusion 109

5 SPECTRA: a Markovian Framework for Managing NFR Tradeoffs with Diverse Levels of Observability 111

5.1 Introduction 112

5.2 Background 115

5.2.1 MDP, POMDP, and MOMDP 115

5.2.2 Multi-Reward Partially Observable Markov Decision Process (MR-POMDP) 117

5.3 Multi-Reward Mixed Observable Markov Decision Process (MR-MOMDP) 120

5.3.1 Model 120

5.3.2 Solver 122

5.4 Motivating Scenario 127

5.5 The SPECTRA Framework 128

5.5.1 Phase 1. NFR Specification 129

5.5.2 Phase 2. Observability Identification 130

5.5.3 Phase 3. Priority Elicitation 131

5.5.4 Phase 4. Targets and Constraints 132

5.5.5 Phase 5. Model Formulation 134

5.5.6 Phase 6. Policy Planning 140

5.6 Evaluations 141

5.6.1 Experiment Setup 142

5.6.2 Evaluation Metrics 144

5.6.3 Baselines 144

5.6.4 (RQ5.1) Optimal Policy Planning 145

5.6.5 (RQ5.2) Policy Quality and NFR Satisfaction 148

5.7 Software Implementation 154

5.8 Threats to Validity 154

5.9 Positioning of The Study 156

5.10 Conclusion 157

6 Exploring SPECTRA within the Context of HitLCPS 158

6.1 Introduction 159
6.2 Case Study: PayNet, a Credit Card Payment System 160
6.3 SPECTRA Framework for PayNet 161
6.3.1 Phase 1. NFR specification 161
6.3.2 Phase 2. Observability Identification 162
6.3.3 Phase 3. Priority Elicitation 163
6.3.4 Phase 4. Targets and Constraints 163
6.3.5 Phase 5. Model Formulation 164
6.3.6 Phase 6. Policy Planning 170
6.4 Evaluation 170
6.4.1 Experiment Setup 170
6.4.2 Evaluation Metrics 172
6.4.3 Baselines 173
6.4.4 Results and Discussion 175
6.5 Threats to Validity 182
6.5.1 Threats to External Validity 182
6.5.2 Threats to Internal Validity 182
6.6 Conclusion 183

7 Appraisal, Concluding Remarks, and Future Directions 184

7.1 Adressing Research Questions (RQ) 184
7.1.1 Research Question 1 185
7.1.2 Research Question 2 186
7.1.3 Research Question 3 187
7.2 Reflections on the Research 188
7.2.1 Feasibility of the Proposed Solutions 188
7.2.2 Digital Twins 190
7.2.3 Computational Overhead 192

7.3 Implications and Future Directions 194

7.3.1 Extension of SOA-HitLCPS ontology 194

7.3.2 Considering Fairness and other HitLCPS Dynamics 195

7.3.3 Improving Policy Planning of SPECTRA 196

7.4 Concluding Remarks 197

Bibliography **199**

Appendix Material **229**

A Classification of Digital Twin Applications in HitLCPS Based on Taxonomy **229**

List of Figures

1.1 Thesis roadmap 9

2.1 Conceptual model of a CPS as a self-adaptive system. Adapted from [1] 15

2.2 MAPE Master/Slave Pattern. 18

2.3 Self-Adaptation Taxonomy [2] 20

2.4 Role allocation in various information processing behaviours and levels of uncertainty. Adapted from [3] 23

2.5 Taxonomy of HitLCPS applications. Adapted with modification from [4] 24

2.6 Service-oriented view of HitLCPS 26

2.7 MAPE-K Self-Adaptive Systems (left) vs. Digital Twin Architectures (right). Images of the robots and their models are taken from [5] . . . 29

2.8 A taxonomy of Digital Twin for Human-in-the-Loop CPS 30

2.9 Various types of Digital Twin 34

3.1 The O*NET content model [6] 53

3.2 Human-machine relationship in HitLCPS 56

3.3 Upper layer service ontology 57

3.4 Service profile 58

3.5 Human capability model 59

3.6 Generic SOA model for self-adaptive systems using MAPE-K hierarchical control pattern 62

3.7 Individuals and classes relationships for Scenario 1. 64

3.8 Ontology model for Scenario 1. 65

3.9 Individuals and classes relationships for Scenario 2. 68

3.10 The subsumption relationships with their Ontoclean metaproperties . 73

4.1 (a) Each task t_j requires worker w_i to move to the pickup location and then to the dropoff location. Every worker is only available to perform tasks within their maximum travel distance $w_i.m$. (b) The bipartite graph matching $G = (W, T, E)$ 90

4.2 Digital twin architecture for heterogeneous spatial crowdsourcing. . . 94

4.3 Instantiation of the digital twin architecture for assisting the design phase. 100

4.4 Fairness evaluation employing 100 workers under different incentive scenarios: IS1, IS2, and IS3. 101

4.5 Unassigned tasks in different incentive scenarios: IS1, IS2, and IS3. . 103

4.6 Instantiation of the digital twin architecture for runtime adaptation. . 105

4.7 (a) Overall unfairness in four datasets with dynamic batch size using scenario IS3 (b) Knee point selection 106

5.1 MR-MDP, MR-POMDP, and MR-MOMDP model 117

5.2 The proposed SPECTRA Framework 129

5.3 The block diagram of MirrorNet as a closed loop control system. . . . 135

5.4 MirrorNet RDM as an MR-MOMDP problem. 136

5.5 Policy planning with OLSAR & OCPerseus. 141

5.6 Digital twins architecture used in the evaluation with RDMSim and MR-MOMDP. 143

5.7 Planning time of MR-POMDP and MR-MOMDP in different settings 146

5.8 CCS of (a) MR-POMDP (164 points) and (b) MR-MOMDP (34 points) under SL2 context. 147

5.9 Expected total reward of MR-MOMDP and MR-POMDP in different environment models. 149

5.10 Expected Total Reward of Scalarised MR-MOMDP and MR-POMDP using SARSOP 150

5.11 Proportions of topology under different conditions. 151

5.12 Target satisfaction under various conditions. 152

5.13 Constraint violation under various conditions. 153

6.1 PayNet as an MR-MOMDP problem. 165

6.2 Digital twins architecture used in the evaluation with MultiMAuS and MR-MOMDP. 171

6.3 Total net revenue gained by each built-in authentication method of MuliMAuS 174

6.4 Total gross revenue on various authentication methods 177

6.5 Total loss on various authentication methods 178

6.6 Total net revenue on various authentication methods 179

6.7 Action proportion in different authentication methods and different random seeds 180

6.8 Constraint violation of Maximise Customer Satisfaction (MAXS) . . . 181

6.9 Constraint violation of Minimise Fraud (MINF) 181

List of Tables

2.1 Fitts' [7] comparison on human versus machine performance, as presented in [3] 23

3.1 Comparison of our SOA-HitLCPS model with other models in answering Competency Questions 71

3.2 Class/Relation Ratio (CRR) Comparison 72

4.1 Worker characteristic 82

4.2 Description of symbols. 86

4.3 Synthetic dataset generation settings and parameters 97

4.4 The comparison of existing fairness-aware task-allocation approaches in spatial crowdsourcing. 98

4.5 Average earnings per worker on different Type B dataset 108

5.1 Phase 1 - NFRs Definition 130

5.2 Phase 2 - Observability Identification 131

5.3 Phase 4 - Targets in environment SL1 and SL2. 133

5.4 Phase 4 - Constraints for normal and detrimental conditions of environment SL1. 133

5.5 Phase 4. Constraints for normal and detrimental conditions of environment SL2. 134

5.6 Fully and partially observable states of MirrorNet. 136

5.7 NFR satisfaction probabilities. 137

5.8 Transition probabilities to fully observable states (Tx) 137

5.9 Transition probabilities to partially observable states (Ty) 138

5.10 Reward vector values for NFRs in MirrorNet. 139

5.11 Observation probabilities of MirrorNet. 139

5.12 Selected optimal weight vectors. 141

5.13 RDMSim configuration 143

5.14 CCS size of MR-MOMDP and MR-POMDP in different environments and settings 147

5.15 Comparison of our proposal with existing approaches 156

6.1 Phase 1 - NFR Definitions 162

6.2 162

6.3 Targets of PayNet 164

6.4 Constraints of PayNet 164

6.5 States in PayNet 166

6.6 Transition function (T_x) of PayNet 167

6.7 Transition function (T_y) of PayNet 167

6.8 Reward function of PayNet 168

6.9 Observation probabilities in PayNet 169

6.10 Configuration parameters for MultiMAuS 171

6.11 Accuracy of each authentication method under different random seeds 176

A.1 Classifying digital twins by human modes, temporal integration, deployment, and maturity level 229

A.2 Classifying digital twins by twinning type, role of human, and feedback. 230

Chapter 1

Introduction

1.1 Motivation

The increasing availability and ongoing integration of enabling technologies have led to the evolution of Cyber-physical systems (CPS) from traditional embedded systems to modern large-scale distributed systems [8]. This, therefore, implies that CPS has all the complexities usually associated with modern large-scale distributed systems. CPS must be able to handle uncertainty and change during operation, control their emergent behaviour, and be scalable and resistant to threats [8].

Engineering self-adaptation in CPS is a challenging task because of the complex, dynamic, uncertain, and resource-constrained nature of this system [9]. Humans have been actively or passively involved in CPS [10]. In the Human-in-the-Loop CPS (HitLCPS), humans participate in the control, decision-making, or monitoring processes along with the automated components [11]. However, the presence of humans in the HitLCPS also raises concerns and requirements related to safety, ethics, and values [12].

Recent works have promoted human values in software engineering [13, 14, 15, 16]. Human values are life goals that indicate what matters to an individual [13] and serve as the principles that guide decision-making at the individual, group, and organisational levels [17]. However, human values have not been given the attention they deserve. Specifically, no established software engineering procedures provide guidelines on defining, refining, and monitoring human values during the software development process [16], in particular self-adaptive systems. Failure to account for human values in software systems can lead to user dissatisfaction and negative socioeconomic consequences [18].

Meanwhile, recent advances in computing and the emergence of enabling technologies have been shifting Industry 4.0 to Industry 5.0 and developing various ways for humans to be involved in human-machine collaboration. One form of human-machine collaboration in CPS is human-machine service provisioning, where humans and machines collaborate to deliver services. Industry 5.0 focuses on three interconnected core values: human-centricity, sustainability, and resilience [19]. It is not technology-driven, as in Industry 4.0, but more as a value-driven initiative that drives technological transformation with a particular purpose [19]. CPS is one of the underlying technologies of Industry 5.0 where human-machine interaction and collaboration can be established, supported by digital twins and simulation [20]. Digital twins have been widely explored to mediate human-machine interaction by handling it virtually or emulating possible modes of interaction [21].

Motivated by these facts, this thesis aims to contribute to a model and framework for self-adaptation in HitLCPS leveraging digital twins as part of the methodology and medium for evaluating the proposed adaptation mechanisms. We begin by examining the role of humans in HitLCPS and identifying values and attributes that impact these roles to build a reference model for HitLCPS. Additionally, we explore fairness as a value that can enhance worker performance in human-machine collaboration. Lastly, we propose a framework for self-adaptation that can be utilised in various domains, taking into account the different nature of the observability of NFRs in HitLCPS. We use digital twins to enable self-adaptation and evaluate our proposals' effectiveness on different scales by simulating various scenarios.

1.2 Problem Statement

In HitLCPS, humans and machines work together to achieve shared objectives. Human roles are diverse, ranging from decision makers (e.g., operators, users, supervisors) to executors (e.g., monitored workers). Understanding the unique attributes of humans and machines is essential for creating an effective partnership.

Many sociological studies [22, 23] highlight that human motivation is individualistic and is influenced by many other factors, including but not limited to human values (e.g., prestige, social justice, etc.). Furthermore, the ethical aspects are the key elements that govern the operation of HitLCPS [11], which is closely linked to human values [17]. Extracted as NFRs, human values should guide the adaptation process to ensure their fulfilment [13].

NFRs play essential roles in ensuring that the system not only performs its intended functions but also in a safe, reliable, efficient way and satisfies other users' requirements (e.g., human values). However, satisfying all NFRs is challenging because they are influenced by several factors that can be unpredictable, such as human behaviour, environmental conditions, and system variability, which all introduce uncertainty into the system's operation.

In HitLCPS, uncertainty can stem from limited knowledge or an incomplete understanding of the system (i.e., epistemic), as well as from the inherent variability or randomness within it (i.e., aleatory) [24]. Therefore, developing a comprehensive model is essential for reducing uncertainty. This model enhances understanding of the system and accounts for the inherent randomness in environmental factors, enabling the system to make robust decisions and operate effectively despite variability.

In this regard, the goal of this thesis is to address the following problems present in the existing works:

Problem 1. Existing literature only covers the limited roles of humans in HitLCPS by treating humans only as service consumers. Therefore, a more comprehensive model in the Everything-as-a-Service paradigm that covers diverse roles of humans and considers the unique characteristics of humans and machines is necessary.

Problem 2. Human values (sc., fairness) are necessary to consider in human-machine service provisioning within HitLCPS. Nevertheless, existing task allocation strategies that promote fairness primarily focus on workers with similar characteristics (i.e., homogeneous). Novel approaches are required to cater to fairness in the increasing prevalence of heterogeneous worker fleets involving autonomous robots/vehicles and humans as service providers.

Problem 3. HitLCPS often has mixed-observable NFRs, where some NFRs are fully observable while others are partially observable due to sensors' limitations or limited available data to process (related to human/human values—e.g., user satisfaction, emotion, etc.—or system-related—e.g., reliability, security, etc.). Current approaches for addressing self-adaptation in HitLCPS to satisfy NFRs amidst uncertainties only consider a uniform level of observability of the NFRs, either fully observable or partially observable; this leads to suboptimal performance and trade-offs in environments with a mixed-observability level of NFRs, necessitating a new approach.

1.3 Research Questions

This thesis addresses the following research questions:

RQ1. How can human-machine collaboration in CPS be engineered for self-adaptation that involves support from humans and machines? What human aspects/properties should be considered?

RQ2. How can human-machine service provisioning be continuously adapted to remain optimal considering human values (sc., fairness) and the constraints of humans and machines?

RQ3. Given that uncertainties can potentially breach the NFRs of human-machine services and necessitate a trade-off, how can these uncertainties be anticipated and the trade-offs dynamically optimised during service provisioning?

1.4 Thesis Objectives and Contributions

This thesis aims to provide a reference model and a framework for self-adaptation in HitLCPS that considers human values and attributes in response to the problems identified and research questions formulated.

The thesis contributes to the field of software engineering concerning HitLCPS.

Specifically, it focuses on advancing the understanding of engineering self-adaptation mechanisms that consider human value, preferences, and uncertainties in HitLCPS; utilising digital twin technology to enable self-adaptation and as a medium for evaluation.

The major contributions are the following:

- **The reference model for humans as service providers in HitLCPS:** This study identifies the challenge of integrating Human-as-a-Service in CPS service composition, highlighting its significance. We introduce a reference ontology model for service-oriented architecture (SOA) within HitLCPS and the broader Everything-as-a-Service context. This innovative model encompasses human dynamics, treating humans as collaborative service providers alongside machines and extending traditional service composition paradigms to enable human-machine collaborations.

- **Fairness-aware human-machine service provisioning:** The primary challenge in defining fairness in human-human and human-machine task allocation lies in reaching a consensus among stakeholders on what constitutes fairness. Additionally, heterogeneous capabilities necessitate a novel approach, as existing fairness frameworks typically focus on homogeneous capabilities. We develop a novel task allocation for heterogeneous human-machine collaboration in a competitive economy involving a diverse fleet of workers, spanning both human-human and human-machine interactions, by explicitly considering three distributive justice principles (i.e., equity, equality, and need). Equity, equality, and need are defined to conceptualise fairness, and we adopt widely used indices to formulate corresponding metrics to quantify unfairness. We use the Gini coefficient to measure inequality, the coefficient of Variation to measure inequity, and Jain's fairness index to measure need-unfairness. A reference architecture of digital twins for heterogeneous crowdsourced logistics is introduced to find balanced tradeoffs between equity, equality, and need.

- **The Markovian framework for managing NFRs tradeoff under uncertainties:** We present a novel framework based on Markov models that surpasses existing work by addressing the heterogeneous observability of NFRs while managing conflicts under uncertainty. This framework acts as a guide for self-adaptive system (SAS) designers, aiding their decision-making based on NFRs metrics observability and uncertainty considerations. To support this framework, we also introduce the Multi-Reward MOMDP (MR-MOMDP), which advances the traditional MOMDP model by accommodating multiple objectives in a mixed-observability environment using vector rewards.

1.5 Publications

The contents of this thesis have been partially or completely derived from the papers published/communicated during the PhD, as listed below. This thesis should be considered the ultimate reference for the details and ideas presented in these publications.

- Published

    1. **A Conceptual Reference Model for Human as a Service Provider in Cyber Physical Systems**
    H.T.N. Ignatius, R. Bahsoon
    2021 IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), In conjunction with 43rd International Conference on Software Engineering (ICSE'21). Publica-

tion date: 29 June 2021.
https://doi.org/10.1109/SEAMS51251.2021.00012

2. **Equity, Equality, and Need: Digital Twins Approach for Fairness-Aware Task Assignment of Heterogeneous Crowdsourced Logistics**
   H.T.N. Ignatius, R. Bahsoon
   IEEE Transactions on Computational Social Systems (TCSS). Publication date: 13 October 2023
   https://doi.org/10.1109/TCSS.2023.3321940

- Communicated

1. **SPECTRA: a Markovian Framework for Managing Non-Functional Requirements Tradeoffs with Diverse Levels of Observability in Self-Adaptive Software Systems**
   H.T.N. Ignatius, H. Samin, R. Bahsoon, N. Bencomo
   ACM Transactions on Autonomous and Adaptive Systems (TAAS) ***(Under review)***

1.6 Thesis Roadmap and Storyline

Figure 1.1 illustrates the thesis roadmap and storyline as follows:

- **Chapter 2. Self-Adaptation and Digital Twins for HitLCPS** This chapter contains a survey of the current literature and state-of-the-art approaches to identify gaps in the HitLCPS domain, specifically in cognition/capability, motivation, and predictability. We also explore how digital twins have been developed for HitLCPS, discuss how digital twins can enable self-adaptation in HitLCPS, and provide a taxonomy.

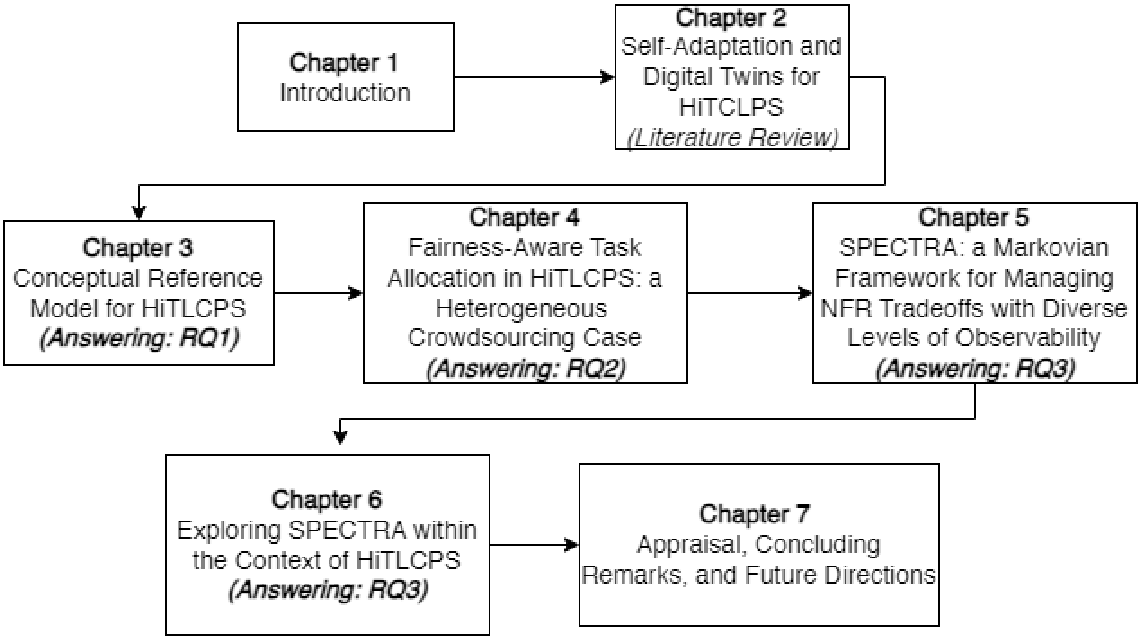


Figure 1.1: Thesis roadmap

- **Chapter 3. Conceptual Reference Model for HitLCPS** In Chapter 2, we highlight the importance of anticipating differences in human and machine capabilities to enable collaboration between humans and machines in service composition. This chapter presents a classification of human-as-a-service in CPS and introduces a Service Oriented Architecture (SOA) ontology model for the HitLCPS environment. The model considers humans' unique characteristics and dynamics as service providers or collaborators adopted from O*NET Framework [6]. To demonstrate the practicality of our conceptual reference model, we provide a use case from the medical domain and implement the reference model as a semantic information model following OWL-S [25] and using Protégé [26] editor. We evaluate the ontological contribution of our model based on criteria such as accuracy, completeness, adaptability, clarity, and consistency. Portions of this chapter have been published in [27].

- **Chapter 4. Fairness-Aware Task Allocation in HitLCPS: a Heterogenous Crowdsourcing Case**

  Based on Chapter 3, we comprehend that human capability is influenced by various characteristics and values of individuals, and one prominent factor is fairness. Numerous studies in the field of social economics indicate a positive correlation between fairness and job satisfaction. In conventional workplaces, fairness is widely recognised as a driving force behind human motivation, loyalty, and productive collaboration. However, contemporary fairness-aware task allocation approaches have mostly focused on homogeneous workers, with equity or equality being the sole fairness requirement. With the rising trend of diverse worker fleets consisting of autonomous robots/vehicles and human-in-the-loop as service providers (e.g., crowdsourced logistics), novel approaches are necessary [28].

In this chapter, we introduce our proposed fairness-aware task allocation strategy for heterogeneous workers that takes into account equity, equality, and need. We accomplish this by applying our principles to the maximum-weight bipartite matching algorithm and evolutionary approach implemented on a digital twin to enable runtime adaptation. We also utilise digital twins to gain a comprehensive understanding of the system's performance by examining various incentive scenarios. We evaluate the overall unfairness, workers' income, and the quantity of expired tasks to determine how effective our approach is. The content of this chapter is derived from our earlier work published in [28].

- **Chapter 5. SPECTRA: a Markovian Framework for Managing NFR Trade-offs with Diverse Levels of Observability**

  In Chapter 4, we employed three fairness principles specified as NFRs with directly measurable metrics/indexes. However, while many NFRs can be directly and fully observed, some are only partially observable. Existing solutions typically consider homogeneous observability, with all the NFRs under investigation being either fully or partially observable. Moreover, existing solutions tend to adopt a single objective approach with scalar rewards, which conceals the individual priorities of each NFR.

This chapter introduces SPECTRA as a framework rooted in the multi-objective Markov decision process (MDP) [29, 30], employing vector rewards to explicitly portray multiple Non-Functional Requirements (NFRs) and their prioritisation during decision-making. As a companion to SPECTRA, we introduce MR-MOMDP, an MDP model with vector reward for mixed-observability environments. SPECTRA is purposefully designed to navigate trade-offs among NFRs within dynamic settings encompassing varying NFR observability. The chapter illustrates how mixed-observable NFRs within a remote data mirroring system can be modelled during the design phase and optimised at runtime using a digital twin architecture. Our approach is evaluated by quantifying the expected total reward, size of the Convex Coverage Sets (CCS), target satisfaction level, and the number of constraint violations. The content of this chapter has been submitted to a journal as [31].

- **Chapter 6. Exploring SPECTRA within the Context of HitLCPS**

  To complement Chapter 5, this chapter utilises a case study of a credit card payment system, illustrating a scenario of HitLCPS where humans function as service consumers. The evaluation is conducted through a digital twin architecture extending MultiMAUS [32], a specialised simulator designed for online credit card transactions with multi-modal authentication features. In this system, users assume the role of service customers, encompassing both genuine and fraudulent customers.

  We introduce a novel heuristic authentication approach for MultiMAuS, dubbed as "Dynamic" and showcase the utilisation of the SPECTRA framework for specifying NFRs and facilitating self-adaptive authentication. Test outcomes indicate that SPECTRA surpasses built-in methods by offering superior trade-offs, notably enhancing the performance of Dynamic, particularly in scenarios characterised by a higher frequency of fraudulent activities.

- **Chapter 7. Appraisal, Concluding Remarks, and Future Directions**

  This chapter evaluates the thesis in general by explaining how research questions were addressed, along with reflections on research pertaining to computational overhead, the feasibility of the proposed solutions, and digital twins. Finally, this chapter concludes the thesis by summarising the main contributions and discussing potential research directions.

Chapter 2

Self-Adaptation and Digital Twins for HitLCPS

HitLCPS is a self-adaptive system dealing with dynamic environments and uncertainties from various sources. Humans play essential roles in HitLCPS, which has some additional challenges compared to traditional CPS. Meanwhile, the digital twin has emerged as a pivotal technology crucial for capturing the dynamic and complex nature of CPS environments. Digital twins have been explored as new modalities to enable self-adaptation [33] as a controller (i.e., autonomous manager, managing system) or as an extra layer to allow lifelong adaptation [34] by simulating and executing what-if scenarios without affecting the real asset (i.e., physical system). This thesis demonstrates the use of digital twins to enable self-adaptation in several domains using various techniques and methods to overcome problems that we identified from the literature.

This chapter provides background information on self-adaptation, CPS, human-in-the-loop, digital twins, and their interrelationships. We discuss how digital twins can help with the self-adaptation of HitLCPS and present a taxonomy of digital twins for HitLCPS. It also includes a literature review to identify gaps in the HitLCPS *domain, focusing on cognition/capability, motivation, and predictability.*

2.1 Self-Adaptive Cyber-Physical Systems

We begin this chapter by discussing the basic concepts of CPS and self-adaptation, including definitions, a conceptual model, and adaptation properties and patterns that can be applied.

2.1.1 Definitions

National Science Foundation (NSF) defines CPS as "engineered systems that are built from and depend upon the seamless integration of computation and physical components" [35]. CPS refers to the integration of computation, networking, and physical processes. Embedded computers and network monitors in cyberspace control the resources and processes in the physical space, such as sensors, robots, and vehicles. There are feedback loops between the physical space conditions and computing in cyberspace, which affect each other [36]. CPS are employed in many aspects of our lives, from communication to transportation. As feedback systems, they require some form of intelligence to take adaptive and predictive actions to enhance their service quality.

CPS is a self-adaptive system by nature. A self-adaptive system (SAS) is a system with the capacity to modify its behaviour based on its understanding of its environment and internal state [37]. It independently determines how to adjust or restructure itself to accommodate alterations in its surroundings and conditions with minimum external intervention. SAS effectively manages changes in external conditions, resource availability, workloads, demands, and unforeseen failures [1].

The diagram in Figure 2.1 illustrates how a CPS can function as a SAS. The physical layer represents the external environment that the SAS interacts with, where the effects of adaptation decisions are observed and evaluated. The cyber layer consists of the managing system, managed system, and adaptation goals. The managed system includes application codes that enable the system's domain functionality through sensors and effectors in the physical layer. The managing system includes the adaptation logic that manages one or more adaptation goals [1]. These adaptation goals are the managing system's concerns over the managed system, usually associated with the software quality attributes of the managed system [1].

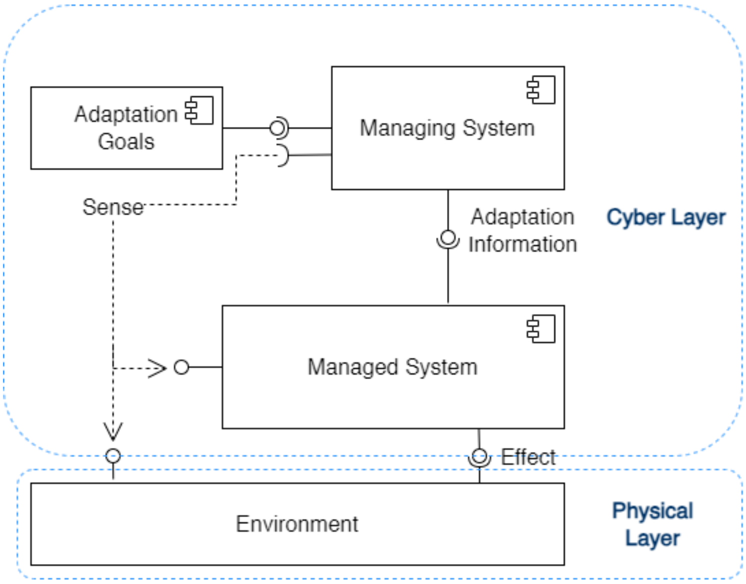


Figure 2.1: Conceptual model of a CPS as a self-adaptive system. Adapted from [1]

2.1.2 Adaptation Properties

When dealing with constantly changing environments, it is crucial for CPS to operate independently, relying less on human involvement and becoming more self-adaptive. This increases their efficiency and guarantees their resilience and effectiveness in responding to any situation.

Self-adaptation is the ability of the system to overcome dynamics and uncertainties. Adaptivity properties are known as self-* properties, which includes:

- *Self-configuring* is the ability to automatically and dynamically reconfigure in response to alterations through the installation, update, integration, and assembly/disassembly of software entities [38].
- *Self-healing* is closely associated with self-diagnosis and self-repair. It involves the capacity to detect, diagnose, and respond to disturbances, as well as the ability to proactively identify potential issues and take appropriate measures to avert failures.[38].
- *Self-optimising*, also known as self-tuning or self-adjustment, involves the ability to oversee performance and resource allocation to satisfy requirements from different users [38].
- *Self-protecting* refers to the ability to identify security breaches and respond to their consequences, encompassing two key dimensions: protecting the system against malicious attacks and proactively anticipating issues to prevent them or reduce their impact.[38].

One property that has been widely discussed in the literature is self-awareness. Some literature views the self-adaptive system as self-aware and vice versa [39]. However, some argue that a self-aware system has distinct characteristics from a self-adaptive system. Chen et al. [40] and Faniyi et al. [41] share a common definition of self-awareness as the ability of the system to obtain knowledge about its current state and the environment. Therefore, a self-aware system can also determine how other components of the system perceive it.

In addition to self-awareness and self-adaptivity, for a comprehensive understanding of the adaptive behaviour of CPS, self-adaptive CPS should also possess a *self-explainability* property [42]. This property ensures that the system can explain and justify its actions and decisions.

This thesis focuses solely on self-adaptive systems, specifically self-optimising systems. We aim to propose a self-optimising method that considers the objectives and NFRs of HitLCPS. Although self-adaptivity requires context information to achieve adaptation, we do not consider our work a self-aware system because we are not concerned about how other components perceive it. Furthermore, our thesis does not use online learning strategies to update its knowledge/model, as [41] suggests for designing self-aware systems.

2.1.3 Adaptation Pattern

Adaptation in CPS is made in a cross-layer that combines different adaptation mechanisms within and across layers, generally using multiple feedback loops [8]. MAPE-K (Monitor-Analyse-Plan-Execute-Knowledge) feedback loop, introduced by IBM [43], has been a prevalent mechanism for self-adaptation adopted in many CPS. The *monitor* component offers mechanisms for gathering, aggregating, filtering, and reporting information from a managed resource/subsystem(s). The *analysis* component includes techniques for correlating, forecasting, and modelling complex situations using the data gathered from the monitoring function. The *plan* component constructs the best plan or actions required to achieve the objectives according to the result of the analysis component and available policies. Lastly, the *execute* component controls the execution of a plan while considering dynamic updates.

The diverse and complex problems of self-adaptive systems require different solutions and architectures. Several patterns of interacting control loops in self-adaptive systems have been identified by [44]. Each pattern shows a unique particular way

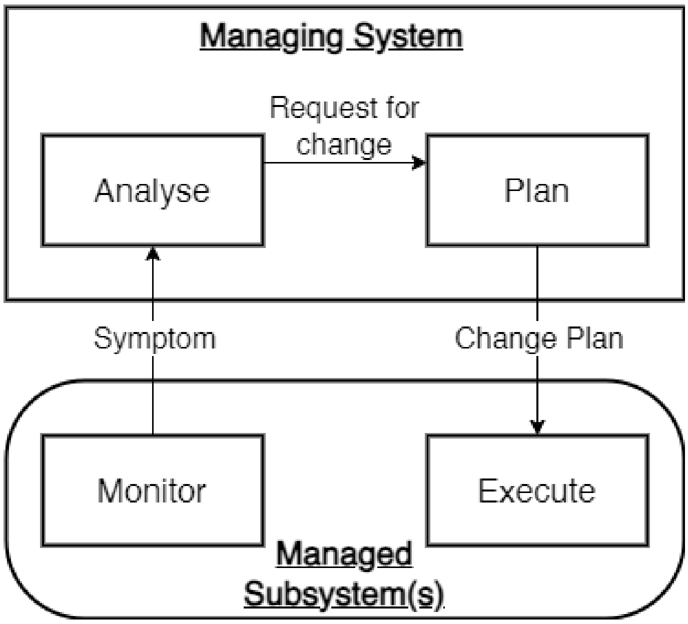


Figure 2.2: MAPE Master/Slave Pattern.

.

to orchestrate the MAPE control loops, as follows:

- *Hierarchical Control* are controlled by a multi-layer hierarchical control structure where complete MAPE loops are present at all hierarchy levels [44]. The higher the layer, the longer the time scale it will operate as it has a more global vision than the lower layer [44].

- *Master/Slave* pattern, as shown in Figure 2.2, presents a hierarchical control where a single master analyses the problem/situation and plans an optimal solution using the data sent from the monitor component at multiple slaves [44]. The plan component coordinates with the execute component on the slaves to implement the solution at the managed subsystems [44].

- *Regional Planner* is where several local hosts are hierarchically related to a single regional host [44]. The local hosts are responsible for monitoring, analysing, and execution, whereas the regional host is responsible for planning and supervising local hosts [44].

- *Fully Decentralised* is where each host implements a complete MAPE loop, whose local M, A, P, and E components coordinate their operation with corresponding peer components of the other hosts [44].

- *Information sharing.* Each host owns MAPE components in this pattern and only shares the M component [44]. Any further action by P and E components is performed by each host regardless of the other hosts [44].

This thesis emphasises hierarchical control and the master/slave pattern, depending on the perspective. From a general CPS perspective, each intelligent node has its own MAPE feedback loop. However, they may be located in different layers with different purposes and objectives. Through a more fine-grain view, i.e., considering only a particular feedback loop, we may observe a master/slave pattern.

To comprehensively address the complexities of self-adaptation in CPS, several primary issues must be tackled, including the coordination of adaptation mechanisms within and across layers and ensuring system-wide consistency of adaptation [8]. According to Zeadally et al. [45], adaptation in CPS can occur at all layers by considering tight coupling to align subsystem behaviour with the overall result. Adaptation in the Cyber layer is carried out concerning CPS behaviour as a whole. Adaptations in this layer are commonly based on MAPE-K and can be done by agents using adaptive control algorithms/models such as Markov or Petri nets [45]. Adaptation in the Physical layer is generally self-adaptive operation behaviour according to a particular context [45]. This research also incorporates MAPE-K at the Cyber layer as a Master, utilising an evolutionary approach and Markov models, as discussed in later chapters. The execution command sent by the Master on the Cyber layer is carried out by the Slave on the Physical layer to adapt its behaviour based on the context.

Krupitzer et al. [2] categorised existing engineering approaches for self-adaptive systems into five dimensions, namely reason, time, technique, level, and adaptation control, as shown in Figure 2.3. Reasons for adaptation can be changes in context, in the system's resources, or triggered by the user (e.g., changing goals). Based on the time dimension, adaptation can be proactive or reactive. Adaptation can use various techniques, such as parameter adaptation (e.g. changing goals, changing models), structure adaptation (e.g. adding/removing additional control loops), or context adaptation. Krupitzer et al. identified different levels of adaptation, namely the application and system software level (cyber), communication (network), and physical (managed resources, context). They divided adaptation control into three categories: an approach, decision criteria, and degree of decentralisation. Referring to the taxonomy in Figure 2.3, in this research, we demonstrate self-adaptation by changing context and resources. We use decision criteria (i.e., models) and adaptation parameters to develop a reactive adaptation approach at the software system level.

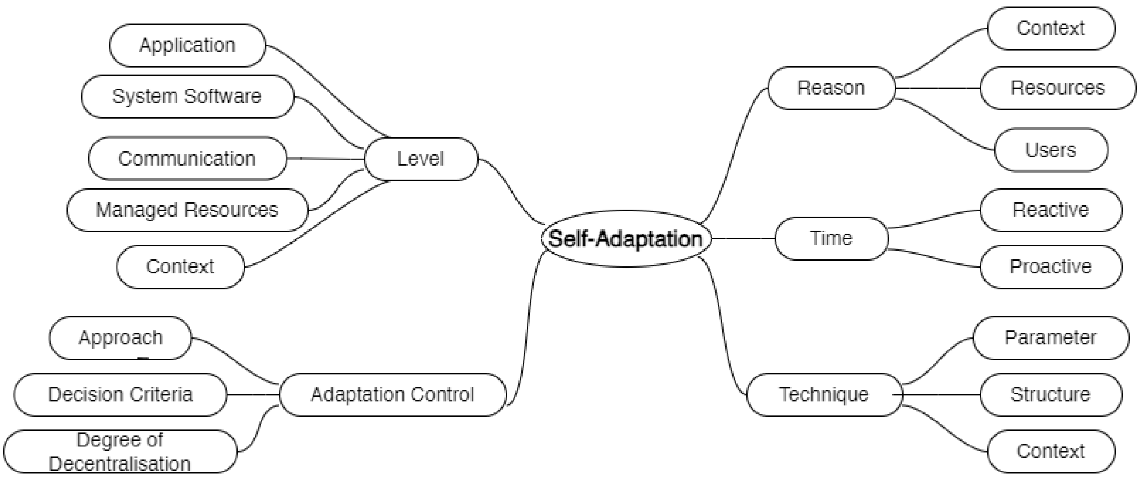


Figure 2.3: Self-Adaptation Taxonomy [2]

.

Musil et al. [46] define three patterns of self-adaptation in CPS, namely *Synthesis-Utilize, Synthesis-Command, Collect-Organize.* In *Synthesis-Utilisation*, information collected from the physical layer is processed and synthesised in the MAPE layer, and the results are utilised by agents (autonomous entities) in the application layer to optimise their service through collaboration. Under the *Synthesis-Command* pattern, the synthesis results on the MAPE layer are commands sent to particular agents to act and configure locally. In *Collect-Organise*, the information collected is used to generate and update global models in the MAPE layer, followed by adjustments by autonomous agents in the application layer using a self-organisation mechanism based on new local and global models. The works offered in this thesis adhere to the Synthesis-Command paradigm for implementing the Master/Slave pattern discussed previously.

2.2 Human-in-the-Loop CPS

The term Human-in-the-Loop has been widely used in the literature, ranging from AI to CPS. Human-in-the-Loop has been defined in AI and ML as the involvement of humans in the learning process and knowledge production [47]. Human-in-the-Loop CPS (HitLCPS) typically consists of a loop involving a human component, cyber component, and physical environment [4]. HitLCPS can also be referred to as Cyber-Physical-Human-System (CPHS) [10] or Cyber-Physical-Social-Systems (CPSS) [48]. Researchers often use these terms interchangeably to describe the interaction between humans and machines in a socio-technical system. Henceforth, we delineate HitLCPS as *a system comprising humans, computing devices, sensors, and actuators intricately interconnected and communicating with one another in pursuit of shared objectives.*

2.2.1 Human vs Machine

Human behaviour is naturally teleological, which changes dynamically influenced by intention, intuition, and experience [49, 50]. Unlike humans, the teleology property of machines or artificial systems emerges due to the programming by the designer. Rasmussen [49] classifies human behaviour into three types of behaviour, namely Skill behaviour, Rule behaviour, and Knowledge behaviour. *Skill-based behaviour* is a sensory-motor activity carried out automatically without conscious control, which is a highly integrated pattern. The next level is *Rule-based behaviour*, which is a composition or work sequence in a familiar environment referring to stored rules or procedures obtained from previous experiences, instructions (e.g. cooking recipes), or the results of previous planning. At the top, we have *Knowledge-based behaviour* that appears when facing unfamiliar situations where there are no know-how, rules, or procedures that can be used, so that performance is more goal-oriented by using several alternative plans and considerations of the consequences.

Building on what Rasmussen proposed, Cummings [3] added another category called *expertise behaviour*. This classification is based on the level of uncertainty experienced in each behaviour. Expertise-based behaviour arises because knowledge is exercised by uncertainties in very different conditions, so that a person may evolve from being knowledge-able to becoming an expert. Uncertainty arises when dealing with many unknown variables, so the situation cannot be fully determined due to a lack of information. It can arise from exogenous sources coming from outside the system, or endogenous sources from within the system.

Cummings suggested task allocations between humans and machines must pay attention to the strengths and weaknesses of each (see Table 2.1). Cummings mapped those four types of information processing behaviours against the degree of automation and uncertainty, as shown in Figure 2.4. Skill-based tasks are the

Attributes	**Machine**	**Human**
Speed	Superior	Reasonably slow
Power Output	Superior	Reasonably weak
Consistency	Ideal for consistent and repetitive tasks	Affected by fatigue and unreliable learning
Information capacity	Multichannel	Primarily single channel
Memory	Ideal for formal, controlled access, and literal reproduction	Better for principles and strategies, access is versatile and innovative
Reasoning computation	Deductive, tedious to program, fast and accurate, poor error correction	Inductive, easier to program, slow, accurate, and good error correction
Sensing	Good at quantitative assessment, poor at pattern recognition	Wide ranges, multifunction, judgment
Perceiving	Copes with variation poorly, susceptible to noise	Copes with variation better, susceptible to noise

Table 2.1: Fitts' [7] comparison on human versus machine performance, as presented in [3]

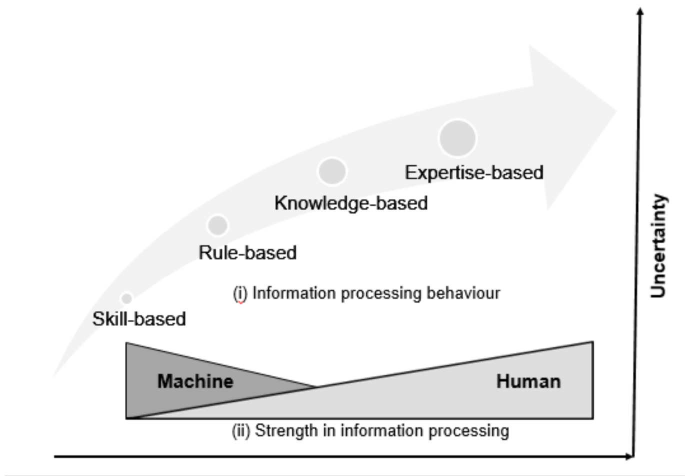


Figure 2.4: Role allocation in various information processing behaviours and levels of uncertainty. Adapted from [3]

best candidates for automation if reliable sensors and feedback support them. If the rule/procedure set is mature and tested, automation can be applied to rule-based tasks. For knowledge-based tasks, automation can be used to help organise and synthesise data. Expertise tasks are tasks with high uncertainties so that expert human reasoning is superior; automation can be used as a tool or partner.

The disparities in capabilities between humans and machines imply structural differences in the capability models within our reference model for HitLCPS, as outlined in Chapter 3.

2.2.2 Taxonomy of HitLCPS

According to Stankovic et al. in [11], it is possible to classify existing HitLCPS applications into three categories, as shown in Figure 2.5, which are (1) Human control, where humans are involved in controlling the system, (2) Human monitoring, where systems passively monitor humans to take appropriate actions, and (3) a hybrid of human control and human monitoring applications.

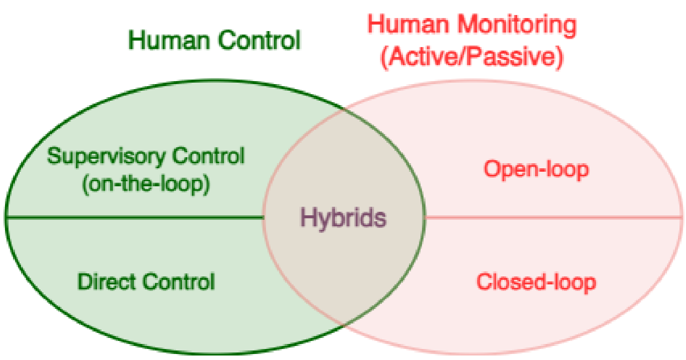


Figure 2.5: Taxonomy of HitLCPS applications. Adapted with modification from [4]

.

Human Control: Nunes et al. [11] provide two main scenarios that can be applied in this category. In a supervisory control scenario, also referred to as Human-on-the-loop [51, 52], the controlling process is mainly autonomous, but human operators are present to oversee it. The operator takes appropriate action to adjust the configuration to help the system make a decision only when necessary. In a direct control scenario, the system relies on humans to provide instructions and report the results back to the human.

Human Monitoring: Many applications commonly monitor humans passively (e.g., in healthcare settings, industrial, transportation, etc.). However, we argue that humans are also capable of actively monitoring themselves or their environment and feeding the system the resulting data (e.g., in crowdsensing systems [53, 54]). We can also further classify this application category into closed-loop and open-loop systems. Open-loop refers to a situation in which the system does not take any action after obtaining the data. Closed-loop systems, in contrast, use the collected data to make decisions towards shared objectives.

Hybrid: Hybrid systems collect data from humans as feedback to their control loops while also considering human inputs in decision-making [11]. One example of a hybrid system could be a crowdsensing system which considers crowd workers' preferences in their task allocation.

This research focuses primarily on humans-in-the-loop, and Chapters 4 and 6 illustrate how passive human monitoring can be implemented as closed-loop systems.

2.2.3 Human roles in HitLCPS

The taxonomy shown in Figure 2.5 and the conceptual model of CPS in Figure 2.1 indicate that humans may be present in every layer. Humans can be part of a managing system that steers the adaptation and provides control; humans can also be part of the managed systems that receive instruction from the managing system; humans can also be part of the environment that is affected by adaptation, and this impact is sensed and communicated back to the upper layers.

Nunes et al. [11] believe that with the support of data acquisition, state inference, and actuation technologies, humans can play various roles in the loop, such as:

- Humans as *sensor nodes* are enabled by wearable devices and ubiquitous sensing technologies provide monitoring data about their mental/physical state or surroundings, actively or passively.
- Humans as *network/communication nodes.* Humans can act as carriers of smart devices with high mobility, which can help the distribution of sensed data and information across the network.
- Humans as *processing nodes*, that can assist in decision making. Given distinct cognitive abilities, humans can complement intelligent agents, especially when dealing with the "known unknown" that requires humans' tacit knowledge to solve [55].
- Humans as *actuators.* With their cognitive abilities and special skills, they can provide assistance, which may require physical action, to help the system accomplish its common goals (e.g., fix the problems when it detects errors).

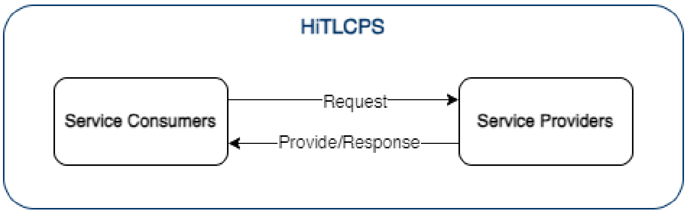


Figure 2.6: Service-oriented view of HitLCPS

.

From the service-oriented perspective, as shown in Figure 2.6, HitLCPS can be seen as a system consisting of service providers and service consumers who help one another to satisfy common objectives. In HitLCPS, humans can be seen as both service consumers and service providers [48]. Humans as *service consumers* use other CPS components to deliver services for them. On the other way around, humans as *service providers* perform appropriate actions that correspond to the requests from the service consumers (e.g., instructions from the managing system/controller component).

Given the limited research on humans as service providers, this study investigates different aspects of humans in this role and how human-machine services can be modelled, composed, and remain optimal in the face of constraints and uncertainties.

2.2.4 Distinguishing Characteristics and Challenges of HitLCPS

Including human components alongside cyber and physical components is a defining characteristic that sets HitLCPS apart from conventional CPS. Sowe et al. [10] emphasise several critical aspects that distinguish a HitLCPS from a traditional CPS, which should be given careful consideration, as follows:

- *Cognition/Capability.* The various methods through which individuals and computers perceive, analyse, act, and respond create challenges and possibilities for collaborative efforts between people and computers to attain a goal effectively [10].

- *Predictability.* The presence of humans in the loop exhibits a new source of uncertainty. Various factors affect human performance over time; humans may lose focus or choose not to follow instructions. However, while humans may be less reliable than machines in following instructions, they have better adaptability in changing environments and use their tacit knowledge to deal with uncertainty.

- *Motivation.* Unlike machines, people require incentives that can take many forms, from monetary incentives to psychological comfort and satisfaction [10]. Motives, skills, and values are important determinants of human performance and action [56]. Values shape and influence the decision-making processes of individuals, groups, and organisations [17]. More than 80 organisations globally have created lists demonstrating their commitment to public value [57]. Fairness is among the five most widely recognised values in these lists [57].

Further, in Section 2.4 we look into various literature and state-of-the-art approaches. We categorise them based on the three aspects mentioned above to highlight the gaps we have identified in each category.

2.3 Digital Twin for Adaptation in HitLCPS

Despite the many definitions that have been put out [58, 59, 60, 61, 62], the basic idea behind digital twins is that they are *a virtual representation of an actual system, real-world entity, or physical asset* [63].

Digital twins are essential for providing system control and representation [64] and are used in experimentation, evolution, design, control, and analysis [63], which are crucial in engineering SAS. Digital twins have emerged as a pivotal technology crucial for capturing the dynamic and complex nature of CPS environments [65].

2.3.1 Digital Twins vs. MAPE-K

The basic architecture of MAPE-K SAS consists of an autonomic manager (i.e., managing system) and the managed element (i.e., managed system/sub-system) [66]. The autonomic manager controls and represents the managed elements, helping humans from the direct responsibility of managing them [66]. Figure 2.7 illustrates how MAPE-K architecture [66] shares several characteristics with digital twin [59]. The autonomic manager corresponds to the digital twin, whereas the managed element is analogous to the physical asset of the digital twin. Furthermore, the autonomic manager and the managed element have bidirectional communication, like a physical asset and its digital twin [59].

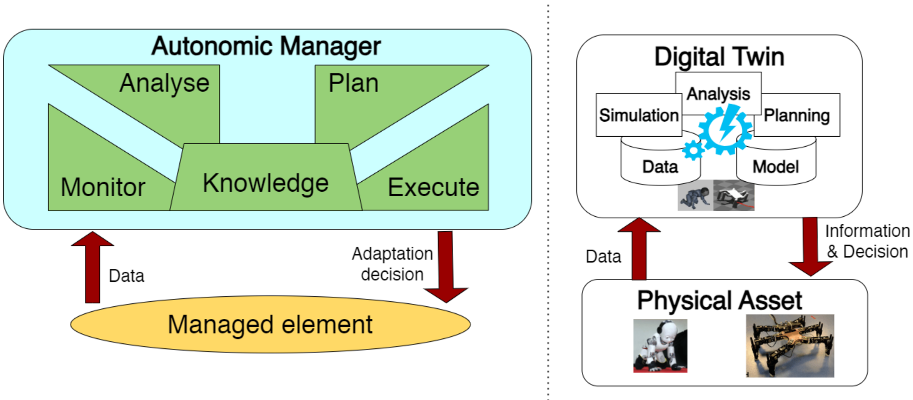


Figure 2.7: MAPE-K Self-Adaptive Systems (left) vs. Digital Twin Architectures (right). Images of the robots and their models are taken from [5]

Digital twins have been studied as new modalities to enable self-adaptation [33] as a controller (i.e., autonomic manager, managing system) [67, 68, 69] or as an additional layer to facilitate lifelong adaptation [34] by simulating and executing what-if scenarios without impacting the real asset (i.e., physical system). The work of [67] extends digital twin to enable continuous optimisation of CPS service delivery and adaptation to changing needs and environments through verification at runtime. In human-robot collaborative assembly [60], digital twins extend virtual simulation models from the design phase to operations for real-time control, task sequencing, and task allocation between humans and robots. The work by [70] uses the digital twin to enable lifelong adaptation for robotic manipulators by predicting future events, updating adaptation conditions, and improving adaptive behaviour.

We have developed a taxonomy for HitLCPS digital twins, shown in Figure 2.8 that categorises them based on deployment, temporal integration, twinning type, feedback type, maturity level, human mode, and role of humans. The taxonomy

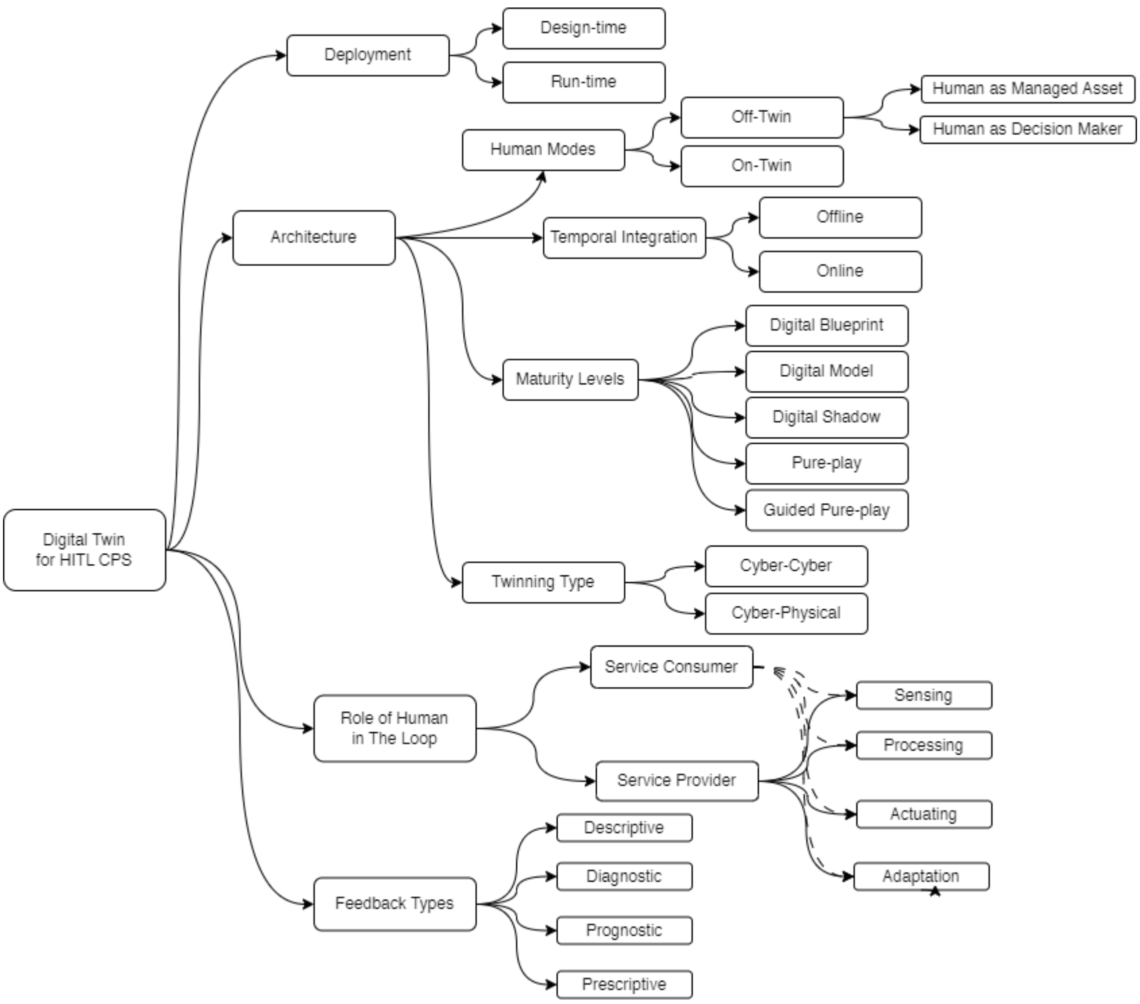


Figure 2.8: A taxonomy of Digital Twin for Human-in-the-Loop CPS

is not exhaustive and should be considered complementary to existing taxonomies [71, 72, 73] that can be expanded and refined as new insights and technologies emerge.

We use the taxonomy to categorise 15 representative articles of digital twins in HitLCPS in Appendix A. The following sections below provide detailed explanations of each category.

2.3.2 Deployment

Digital twins are highly beneficial for developing HitLCPS at the design time and facilitating adaptation during runtime.

- *Design time.* At design time, digital twins are useful for evaluating and exploring various strategies available in the design space. The digital twin can be considered a ”sandbox” where operations can be repeatedly simulated and evaluated, guiding the implementation [74]. The results of this exploration are an optimal blueprint [75] for real asset deployment and optimal strategies/policies that guide the adaptation process of existing real assets. The designers analyse the feedback and oversee and approve synchronisation. The types of digital twins that are suitable for design time are digital blueprints and digital models.

- *Runtime.* At runtime, the digital twin works alongside the asset to facilitate the asset's adaptation, analysis, and maintenance [75]. The digital twin constructs the desired control inputs and generates control instructions by comparing observed and desired behaviour [75]. Synchronisation and feedback from digital twins to assets can be done automatically. Digital shadow, Pure-play, and Guided Pure-play are suitable for application at this stage.

2.3.3 Temporal Integration

The connectivity and temporal integration between digital twins and their assets is one of the differences between digital twins and traditional simulations [76]. It can be classified as follows:

- *Offline.* Offline digital twins (e.g., [77]) are suitable for scenarios where real-time communication is not essential [76]. An offline digital twin would connect to the physical system periodically [76].

- *Online.* Online digital twins are connected to their physical assets in real-time [76] for immediate data exchange and interaction. This connectivity allows the digital twin to be synchronised at a high level of fidelity for analysis and decision-making.

2.3.4 Twinning Type

In CPS, real assets represented by digital twins can be either in the physical layer or the cyber layer.

- *Cyber-Cyber.* Digital twins are virtual replicas of software systems and processes. The cyber-cyber digital twin can adapt completely, corresponding to its twin [78]. Cyber-cyber digital twins themselves can have digital twins, resulting in a hierarchy of digital twins [78].

- *Cyber-Physical.* Digital twins are digital replicas of physical assets, not software systems or processes. The synchronisation is only partial and limited to the configuration and software components of the physical assets [78].

2.3.5 Feedback Type

As a replica, the digital twin shares many features with the assets it represents. The digital twin, on the other hand, includes a number of additional components that enable it to provide simulation and analytic services. We identify several types of planning & analysis functions that can be performed by the digital twins, as follows [58, 79]:

- *Descriptive* - explain the state of assets, monitor their behaviour and detect violations or anomalies by performing descriptive analysis.
- *Diagnostic* - undertake advanced analytics aimed at elucidating why something occurred based on data analysis
- *Prognostic/Predictive* - extrapolate what might happen in the future, use predictive data analytics such as data mining, statistical approaches, and machine learning.
- *Prescriptive* - perform the prescriptive analysis for optimation or mitigation, which allows the system to view probable decisions and follow them through to a predicted outcome based on both current and historical data. Prescriptive analysis uses algorithms, machine learning, and computational modelling to determine the viability of actions before they are done.

2.3.6 Digital Twin Levels

The concept of a digital twin operates on multiple levels, as shown in Figure 2.9, with different levels of maturity and sophistication [63], to facilitate situations where a physical system is unnecessary or when there is no direct match between the internal states of the physical and virtual representations [80].

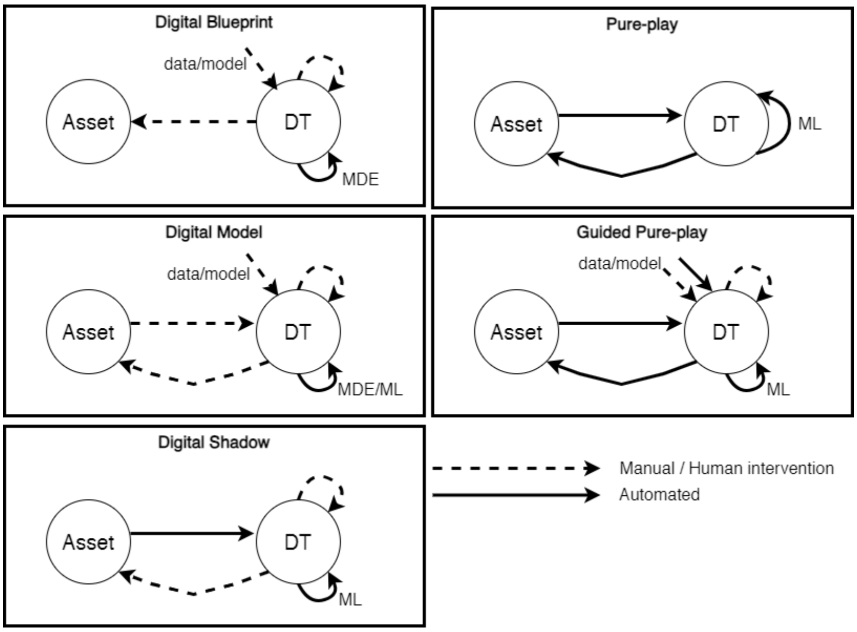


Figure 2.9: Various types of Digital Twin

The following are different maturity levels of digital twins characterised by its seamless connection [80]:

- *Digital Blueprint.* At this level, Digital Twin is only a simulation to build a blueprint for a complex system (i.e., the asset). This type is commonly found at the design stage, where the digital twin is used to explore the design space; hence, a digital twin can be discarded after the design process is complete. Synchronisation between digital twins and assets is done manually because digital twins are not connected to assets. Even analysis can be done using synthetic datasets [80].

- *Digital Model.* The Digital Twin is a simulation that is connected to an asset. The analysis uses historical data produced by and gathered manually from the asset, leveraging ML techniques, and may follow model-driven engineering paradigms and the results configured to the asset manually [80].

- *Digital Shadow*. The Digital Twin runs alongside assets to provide behavioural insights and perform what-if analysis. The Digital Twin collects the data from the asset automatically. The analysis results are consumed by humans and reflected back into the asset manually [80, 81].

- *Pure-Play*. The digital twin is the shadow of the asset that holds the ideal behaviour of the asset. Through extensive use of ML, Digital Twin controls the asset to achieve the system objectives. The synchronisation is done regularly and frequently [80].

- *Guided Pure-Play*. We add this category for a pure-play extension where humans are involved in helping the Digital Twin overcome uncertainty through decision-making in critical and unforeseen conditions and guiding the Digital Twin in anticipating contexts that might occur in the future. This type of digital twin can be implemented on a hybrid HitLCPS, which integrates human control and human monitoring, as discussed in [82].

2.3.7 Layers and Human Modes

There are at least three primary layers of Digital Twin architecture: the *asset layer*, which encompasses real-world systems such as humans, machines, environments, sensors, and other physical systems; the *digital twin layer*, which comprises digital replicas of the entities in the asset layer; and the *decision layer*, which serves as the decision maker [28]. The decision layer is an independent layer because it frequently involves humans or external services/components in making decisions, which occur outside the digital twin layer. Nevertheless, if the decision is made by the digital twin, then this layer merges with the digital twin layer. In Chapter 4 we showcase a three-layered digital twin reference architecture tailored for heterogeneous crowd-sourcing. However, one could use a more fine-grained architecture with more layers to group components into more specific functions, such as in [83, 84, 61].

In the digital twin architecture for HitLCPS, humans can exist in one or more of the three layers. In the *asset layer*, humans are considered a crucial element of the environment that is monitored and managed by the digital twins. Humans are also included in the *digital twins layer* as models for conducting simulations and analysis. Human expertise is of utmost importance at the *decision layer*, as it plays a significant role in making important and strategic decisions.

We classify human presence on the digital twin architecture into two modes: *Off-twin*, where humans operate on layers other than the digital twin layer, such as the decision layer [85, 86], asset layer (e.g., as action executor [62]), or both [62]; *On-twin*, where humans are represented on the digital twin layer [87, 88, 89, 90].

In the off-twin mode, there are two human roles: The first is as managed assets in the asset layer. This means that the digital twin, as the managing system, provides them with instructions; humans are the action executors [62]. The second role is as a decision maker in the decision layer [91, 28, 92, 93], performing analysis and decision-making based on the feedback from the digital twin layer.

2.4 Identified Gaps in HitLCPS

The integration of human elements into CPS research has revealed numerous challenges. Specifically, human dynamics introduces additional complexity that remains unexplored [94]. To optimise the system's performance and minimise errors, HitLCPS must accurately identify the capabilities of humans and machines. Monitoring their unique characteristics and abilities is crucial, as they may change over time. However, these traits, other human attributes, and mental states can be challenging to observe and add additional uncertainties.

We conducted a literature review for each category that differentiates HitLCPS from traditional CPS, referring to Sowe et al. [10], namely cognition (i.e., capability), motivation, and predictability, to identify the gaps that we aim to address in this study.

2.4.1 Cognition/Capability

HitLCPS requires interoperability between heterogeneous components for human-machine collaboration with different capabilities. Service-oriented architecture (SOA) is widely adopted as an underlying technology for CPS [95], offering interoperability through open standards and protocols, flexibility, service abstraction, and scalability.

Ontology plays a significant role in SOA by providing a formal and structured way to understand service semantics, support interoperability, service discovery, service composition, and adaptation. Therefore, having a structured model, e.g., ontology, that captures distinctive cognitive mechanisms and capabilities of humans and machines is essential for HitLCPS.

Huang et al. [96] presented a Physical-Entity service-oriented architecture model to enable inter-operation and coordinated sharing of distributed and heterogeneous data using Physical-Entity (PE) ontology that classifies the physical entities (including humans) and their class properties and services. The proposed Service Model follows the OWL-S model, including the Input, Output, Precondition, and Effect specifications. However, this model divides precondition and effect into context precondition, non-context precondition, non-context effect, and context effect. The non-context effect is a change in the world or environment after service execution.

Meanwhile, the context effect changes the service provider entity after performing the service. There are service provision constraints that represent the physical constraint of the PE relevant to the service, such as maximum distance, maximum load, etc. However, this model has not accommodated human characteristics and capabilities that affect human service quality.

Echoed [96], Wang et al. [97] proposed an ontology model for the context-sensitive specification of the service abilities of physical entities. Physical Entities (PE) provide atomic services with behaviour constraints such as context precondition, precondition, and postcondition. Context preconditions correspond to preconditions related to the dynamic context of the PE that should be established before PE can provide the services. The precondition accounts for the service constraints irrelevant to the context, whereas the postcondition is the condition after a service's execution. However, this model does not adequately accommodate humans as a service because humans can provide composite services in addition to atomic services.

Zhu et al. [98] extended OWL-S ontology concerning several significant issues related to CPS and IoT where every Physical Thing (PT) entity can provide a service and receive the impacts from any service. PT entities are described in four main classes: "Physical Profile", "Operation Profile", "Operation Schedule", and "Context". Zhu et al. introduced the "AppliedTo" concept to the "Service model" to simplify the reasoning process. The "AppliedTo" class represents the recipient PT and effects that can change the recipient PT's state after the service's execution. This model does not consider humans as part of PT but considers humans as part of the environment, acting as service consumers.

Sun et al. [99] proposed an ontology-based CPS service model with location, physical entities, and CPS services as the three main components in which the

CPS service uses physical entities and has effects on them. This model pays more attention to the location and state of physical entities, regarded as context. Several characteristics for physical entities were introduced, such as operation space (working region), degree of parallelism (whether this physical entity can be used by more than several services simultaneously), and working state (the availability state of the physical entity). However, their model does not consider humans in the loop. Additionally, they separate physical entities from the CPS, which does not fit our definition of HitLCPS.

A human service capability description (HSCD) model has been proposed by Sowe et al. [10]. The main objectives of this model are to represent the person's identity, the tasks a person can perform, the qualifications of the person for performing the tasks, and the types of interfaces that can be used to interact with the person. This model mainly focuses on humans and fails to explain how humans' and machines' capabilities can be orchestrated in specific environmental contexts.

Our findings suggest that even though researchers and designers understand cognition and capability differences between humans and machines, existing literature has insufficiently modelled them. A more comprehensive model of human-machine capability is required to serve as a foundational framework for designing, managing, and optimising collaborative work between humans and machines in HitLCPS that promotes efficiency, reliability, interoperability, and adaptability. A more comprehensive ontology provides more detailed knowledge in a particular domain, enhancing usefulness for specific tasks. However, this detail often comes at the cost of generality. Designing the ontology requires careful consideration to ensure it is adaptable based on context [100]. This allows the extension of the ontology to specific details when necessary while remaining broad otherwise [101].

Gap 1. The necessity of a more comprehensive ontology and capability model for human-machine collaboration, addressed in Chapter 3.

2.4.2 Motivation

Wu et al. [102] emphasise the role of fairness perceptions in motivating employees. They suggest that enhancing fairness can increase motivation and improve work performance. Additionally, their perceptions of fairness influence employees' attitudes and behaviours towards work, ultimately affecting their motivation.

One instance of HitLCPS is a spatial crowdsourcing system that involves human and machine workers. In recent years, spatial crowdsourcing studies have evolved to include ubiquitous machines as participants, thanks to advancements in robotics and AI [103]. Task-assignment problems for spatial crowdsourcing often involve spatial and temporal constraints [104], as well as additional constraints such as quality [105] and budget constraints [106]. However, fairness considerations are frequently overlooked, including fairness in the distribution of tasks between machines and humans. Below are some studies that prioritise fairness as an objective in their algorithm for task allocation in spatial crowdsourcing.

Basik et al.[107] used the bipartite-graph approach and equity-based distributive fairness by minimising the difference in the local-assignment ratio among workers. This algorithm works in a setting where workers can either decline or accept job offers. The more offers they accept, the higher their chance of being allocated a task. However, there are settings where the server assigns the task, and the workers do not have the right to choose the assignment.

Chen et al. [108] also used a bipartite graph in their proposal, where unfairness (fairness cost) is quantified as the discrepancy between a worker's deserved bonus proportion and their actual allocated proportion. Their approach favours workers who have completed more tasks in previous rounds, leading to a bias towards workers with superior performance in heterogeneous worker settings. Hence, different approaches to ensure fairness for the less advantaged workers are necessary.

Lan et al. [109] proposed two heuristic techniques for allocating multiple tasks, which measure fairness using a fairness index related to the equal distribution of tasks among workers. However, they do not account for varying task rewards or different worker incentive schemes.

Zhao et al. [110] introduced two heuristic algorithms that utilise game theory to minimize the payoff differences among workers as a fairness metric. Worker payoff is defined as the ratio between task rewards and travel time, incentivising longer task completion times. However, in human-machine crowdsourcing with a diverse fleet of workers, it is necessary to consider varying travel speeds and maximum distance.

The aforementioned approaches for fairness in task allocation are suitable for homogeneous crowdsourcing with single-type human workers. However, such approaches are inadequate for heterogeneous settings involving multiple types of workers, including human and machine workers. To ensure fairness in such settings, it is crucial to modify existing approaches and develop new ones that account for the diverse capabilities and limitations of different worker types. Furthermore, with the increasing need for online task assignments, there is an added layer of complexity to the problem, requiring adaptability in a dynamic environment.

Gap 2. A fairness-aware approach to allocating tasks and rewards that considers the distinct attributes of humans and machines is required, addressed in Chapter 4.

2.4.3 Predictability

Uncertainty is a major underlying issue that can arise from a variety of sources, including insufficient knowledge, sensor noise, inacurate understanding, and the uncertain behaviour of humans in the loop [111]. Many aspects of human dynamics affected by internal and external factors remain challenging to predict and difficult to observe.

HitLCPS, as a socio-technical adaptive system, is given objectives through NFRs that need to be satisfied. Sutcliffe et al. [15] propose the concept of "soft requirements" which expands traditional NFRs to cover socio-political and human-oriented issues that may affect system-level requirements. Therefore, It is important for HitLCPS to be able to monitor the satisfaction level of each NFR. Nevertheless, not all NFRs can be directly observed and monitored, some of them, e.g., human mental states, are partially observable.

Introduced by Reinforcement Learning and the robotics community, Markov Decision Processes (MDP) and Partially Observable MDP (POMDP) have been applied to model sequential decision-making processes in various domains of SAS, including SOA-based systems [112, 113, 114, 115], cloud computing [116, 117, 118], mission planning [119, 120], and security [121, 122]. While one may assume that the impact of the adaptation action is deterministic [122], much research based on Markov processes assumes that the effects of adaptation on NFRs are uncertain.

The following works use Markov models for decision-making in their self-adaptation. We divide it into two sections: section Observability, which discusses the existing works from the observability perspective of the problem under investigation, and section Reward Function, which discusses the reward function used.

Observability

0bservability is the degree to which one can comprehend the internal state of a system based only on knowledge of its external outputs [123]. A system can have either fully-observable states, partially-observable states, or mixed-observable states.

The studies conducted by [116, 113, 114, 112, 115, 124, 125, 29] share common objectives with our work in the endeavour of enhancing Quality of Service (QoS) and ensuring the satisfaction of NFRs. The works by [112, 113, 115, 124, 125, 29, 114] utilise state space to represent information related to the satisfaction level of NFRs. However, only RE-STORM [124] and Pri-AwaRE [126, 29] specifically demonstrate how NFRs, which can only be partially observed in the context, are mapped and represented by the states of the POMDP model.

In other studies using MDP [112, 127, 116, 113, 120, 121, 122, 119, 114, 117, 118], the agent works in an environment where it could fully observe the current state. However, in POMDP-based systems [115, 124, 125, 29], the agent can only partially observe the current state of the environment. Adopting POMDP in SASs is less common than MDP for various reasons. Besides additional computational complexity, not all domains or applications require modelling and reasoning under partial observability. It merits attention that all studies mentioned above only focus on one type of observability (i.e., homogeneous/uniform observability), either fully observable or partially observable states.

MOMDP [128] considers mixed observability and solves it by representing separately the fully and partially observable components of an environment's state (i.e., robot) and derives a compact lower-dimensional representation of its belief space, which can be combined with any point-based algorithm to compute approximate POMDP solutions, resulting in significant improvements in performance. Recognising the importance of employing MDP and POMDP in settings characterised by uniform observability, we suggest exploring MOMDP to manage decision-making in mixed observability environments. This aspect remains relatively unexplored, especially within the areas of software engineering and requirements engineering.

Reward Function

The reward function is an incentive mechanism to provide reward/punishment for an action chosen in each state. The agent's goal is to maximise total rewards in the long run.

Existing approaches generally use single-objective approaches with scalar rewards in their models [112, 127, 116, 113, 120, 121, 122, 119, 117, 118, 115, 124, 125]. The works by [116, 112] rely on the reward function to meet NFRs instead of using NFR satisfaction level as a determinant of the environment's current state. Although the model does not explicitly provide information on the satisfaction level of NFRs, the scalar reward function used in their MDP model aims to optimise the overall utility of the system by prioritising essential quality constraints.

While scalar rewards are generally static, ARRoW [129] introduces dynamic rewards representing the preferences/weights over NFRs updated at runtime by monitoring the current levels of satisfaction of NFRs (i.e., states), the satisfaction threshold for each NFR, and comparing it to a set of satisfaction ranges. The weights associated with the NFRs are mapped onto a decision-making specification using a quantitative requirement prioritisation scheme based on the P-CNP method. The resulting priorities are then aggregated into a single scalar reward value, representing the overall importance of each NFR.

One issue with the scalar approach is that it hides important information about the significance of meeting each NFR in various states of the environment [29]. Scalar rewards also obscure the effects of decisions on adapting to specific NFRs. MR-POMDP++ [29] tackles the problem of scalar reward by suggesting an approach based on MR-POMDP [30]. This method employs a reward vector to represent the priorities of objectives (i.e., NFRs) by indicating how desirable they are in terms of satisfaction, given a particular state of the environment.

Inspired by the work of Samin et al. in [29], we use the reward vector and solver they used to be applied onto MOMDP [128]. In addition, we develop a framework to specify NFRs and their observability and to build appropriate Markov models to consider tradeoffs and uncertainties which we discuss in Chapter 5 and 6.

Gap 3. The necessity of a framework and model for managing NFRs tradeoffs to consider various types of observability, addressed in Chapter 5 and 6.

2.5 Summary

We have discussed the concepts of self-adaptation, HitLCPS, and digital twins, emphasising their connections. We have presented a taxonomy of digital twins for HitLCPS, which serves as our foundation for building digital twins in the next chapters of this thesis. Moreover, we have identified gaps in current research, including the necessity for a holistic model for human-machine collaboration that accommodates the distinct traits of both humans and machines. Additionally, fairness must be considered when distributing tasks and rewards among humans and machines, respecting their unique capabilities and skill sets. Lastly, there is a requirement for a framework and model to navigate trade-offs in NFRs, taking into account different types of observability.

Chapter 3

A Conceptual Reference Model for HitLCPS

This chapter provides a classification of human-as-a-service in CPS, and we propose a Service-Oriented Architecture (SOA) ontology model for the CPS environment as part of the Everything-as-a-Service paradigm. The model considers human characteristics and their dynamics, as a service provider or collaborator with the machine. As the ontology model is an enabler for engineering a self-adaptive CPS with human-machine collaboration as service providers, we describe how a commonly used self-adaptive reference model can be refined to benefit from the vision. We evaluate the ontological contribution against criteria that relate to accuracy, completeness, adaptability, clarity, and consistency. We demonstrate the feasibility of our conceptual reference model using use cases from the medical domain, and we show how human-machine service provision is possible. This chapter originally appeared as [27].

3.1 Introduction

HitLCPS has been broadly adopted across various fields. In smart manufacturing, human-machine collaboration provides flexibility that allows manufacturers to adapt more easily to shifting demands in products and processes [130]. Human-machine collaboration is also essential in the space system domain as it enhances operational efficiency, safety, and adaptability while also addressing the unique challenges in remote environments [131]. While fully automated CPS can excel in strength, precision, and speed, humans possess cognitive abilities, consciousness, and skills that allow them to quickly adapt to new requirements and tasks.

As discussed in Chapter 2, humans and machines differ in many aspects. Humans work based on their consciousness, while machines operate based on what is programmed [132]. Humans have to work based on a motive, which is often the result of a trade-off analysis between rewards and risks. Humans have free will, so humans can decide to stop working or choose to do work differently based on the context and their considerations. Besides, many factors influence human performance, such as mood, fatigue, incentives, etc. In many CPS applications, humans, when kept in the loop, are generally an operator; the users who instruct or initiate requests for and receive services from CPSs (*service consumers*). However, many complex CPSs are essentially a combination of computers, machines, and people who work together to achieve system goals [10]. In such systems, humans can provide services by performing tasks based on their ability to sense, act, store, and process data (e.g., citizen sensing, citizen actuation [133]). Therefore, humans in the loop can be viewed as not only the service consumers but also as the *service providers* [48].

Service-Oriented Architecture (SOA) provides potential solutions for modelling, run-time synthesis, management, and composition of HitLCPS to deal with variability in component types and changing application environments at runtime [134]. With SOA, every capability possessed by each entity is considered either an atomic or composite service. However, traditional SOA models and composition techniques have limitations when directly applied to CPS for various reasons due to the heterogeneity of physical entities, whether human or machine, while considering context requirements, service provision constraints, and service similarity.

Several works have proposed service models and service composition for CPS [96, 135, 136, 98], CPSS [134], and systems of systems in general [137, 138]. However, these studies do not pay much attention to humans as service providers in CPS. Human characteristics are not explicitly modelled. Humans are mostly considered part of the physical entities, along with robots, vehicles, sensors, and other actuators. Existing models are not adequate to accommodate humans as service providers and new or enhanced models are needed.

The novel contributions of this chapter are as follows: We first define the problem of Human-as-a-Service in CPS service composition, and we motivate its need. We contribute to a novel reference service-oriented SOA ontology model for the CPS environment as part of the Everything-as-a-Service paradigm. The model considers human characteristics and their dynamics, as a service provider or collaborator with the machine. The model builds on existing service composition paradigms and extends it beyond the machine-centric ones to also include human-as-a-service in CPS. As the ontology model is an enabler and pre-requisite for engineering a self-adaptive CPS with human-machine collaboration as service providers, we describe how a commonly used self-adaptive reference model, MAPE-K can be refined to realise the vision. The proposal is a pragmatic shift towards acknowledging that both humans and machines work in collaboration as service providers. The paradigm can enable new modalities of service composition, where humans can assist the machine (vice-versa is also possible), considering some qualitative attributes such as accumulated experience, knowledge, skill, abilities, and other human attributes such as emotion, mood, compassion, fatigue, etc.

The ultimate vision is to transit the problem of service composition into a collaborative human-machine service composition, where bidirectional infosymbiotic cooperation/learning between the machine and human can be envisioned, promising more dependable and human-centric CPS services provision. We report on how the model can be instantiated using a use case from the medical domain. We follow the standard and commonly used approaches to ontology evaluation, where we evaluate the ontology against criteria that relate to accuracy, completeness, adaptability, clarity, and consistency.

3.2 *-as-a-Service in HitLCPS

3.2.1 Everything as a Service

Everything as a service (XaaS) is a concept for services and applications that users can access over the network, which is generally found in the form of Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). However, in its development, we also see more specific terms such as Communication-as-a-Service (CaaS), which provides VoIP services, Transportation-as-a-Service (TaaS), such as online taxi or ride-hailing services, and many others.

HitLCPS integrates computation, networking, and physical processes that involve humans in the loop. Mobile internet devices with varying computing capabilities are strong candidates for implementation in physical entities of the CPS [139]. Some machines may have limited computing resources, but many devices could execute complex computations and processes. These devices can communicate and share services over the network. The development of ubiquitous computing technology allows human-computer interaction to a higher level with various interfaces. Humans can be accessed and interacted with the system through handheld devices or other human interface devices (HID) nearby. A service-oriented architecture is therefore promising for HitLCPS to enable collaboration between components in providing services.

3.2.2 Human-as-a-Service

The idea of human-as-a-service supports the XaaS (Everything-as-a-Service) paradigm that sees that humans can provide services to the system; so can other devices.

Human-as-a-service in CPS is defined as *a "thing" of Everything as a Service with human capabilities and properties.* These are humans as service providers that can work either in isolation or in collaboration with machines in CPS to sense, process computation, actuate, learn, and/or transfer its learning with the objective of providing more socio-dependable and human-centric service composition models for CPS. The relation can be collaborative or an arms race (as we discuss in Chapter 4), based on the context with the incentive of a better overall service provision.

Human-as-a-service is widely manifested as an individual or group of services, often by direct appointment or through an open-call (crowdsourcing) mechanism. It exhibits unique characteristics as it evolves during its life cycle and involves various ways of collaboration/communication [140, 141].

Human-as-a-service within the CPS can vary. Treating CPS as a self-adaptive system using the MAPE-K [142] reference architecture, human-as-a-service can be applied within all layers of monitoring, analysis, planning, execution, and knowledge. Nunes et al. [11] have identified several human roles in the loop that we use as a reference to categorise human-as-a-service in HitLCPS as follows:

1. *Sensor service*. In their activities, humans might use tools and computing devices equipped with digital sensing functions. Humans also have five natural senses that can detect many events (e.g., traffic hours, car accidents, fires, etc.). Humans can provide this service actively by reporting an event or phenomenon detected by the five senses and passively by allowing their activity/behaviour to be recorded to see social phenomena (social sensors).

2. *Processing service* - Humans are learning creatures who have developed cognitive abilities. With his diverse knowledge and intuition, human choices will help make decisions, especially when dealing with uncertainties due to lack of knowledge or other environmental dynamics.

3. *Actuating service* - In everyday life, humans already act as actuators. When receiving an emergency signal from the patient's room, the nurse will immediately go to the patient's room and take the necessary actions. Within the scope of HitLCPS, sensor networks or robots may detect errors and require specialised actuation from humans to fix the problem [11].

4. *Adaptation Service* - This service is a composite of the three services above. Humans can act as adaptation promoter for other nodes. "The users (with different roles) may decide whether the adaptation is needed, which strategy to choose, and even participate in its realisation" [143]. This role includes, but is not limited to, control feedback, provision of knowledge, learning, and evaluation.

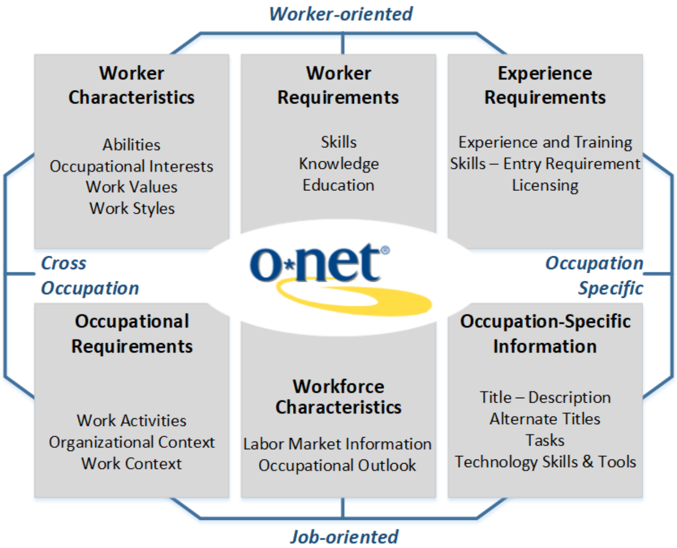


Figure 3.1: The O*NET content model [6]

3.3 The O*NET Framework

Understanding the classification and relationship between the attributes of workers and their jobs is essential to build an adequate human-as-a-service model and prerequisite for developing a self-adaptive model. We have studied several existing frameworks, namely O*NET [6], SOC [144], and ISCO [145]. We have decided to use O*NET as it is considered to be the primary reference for building human capability models.

The Occupational Information Network (O*NET) provides a rich database of occupation information that describes the job and worker characteristics. The Content Model defines the most important types of job information and incorporates them into a theoretical and empirical framework.

In the O*NET framework, every single job requires a different selection of knowledge, skills, and abilities and performs a variety of tasks and activities. These particular characteristics of an occupation are described by the Content Model (as seen in Figure 3.1, which reflects the characters of occupations (job-oriented descriptors) and people (via worker-oriented descriptors).

Worker-oriented descriptors consist of several attributes as follows:

1. Worker Characteristics are defined as enduring features that can affect both performance and the capacity to learn the knowledge and skills necessary for the efficient performance of the job. These characteristics are classified as follows:

   (a) Abilities: enduring attributes of the person that affect performance.

   (b) Occupational Interests: Preferences for conditions/environments at work.

   (c) Work Values: Global aspects of work consist of basic needs that are essential to an individual's satisfaction.

   (d) Work Styles: Personal features that can influence how well someone does work.

2. Worker Requirements reflect an individual's developed or acquired qualities that may be correlated with work performance. These attributes are categorised as follows:

   (a) Skills: Developed capacities that promote learning (faster acquisition of knowledge) and performance of activities that occur across jobs.

   (b) Knowledge: organised sets of concepts and facts applying in general domains.

   (c) Education: Prior academic experience needed to perform in a job.

3. Worker Experience Requirements are previous work experiences that involve employee experiential backgrounds such as certification, licencing, and training.

(a) Experience and Training: relevant work experience, apprenticeship, and on-site/on-the-job training required.

(b) Skills-Entry Requirement: entry requirement for developed capacities that facilitate learning and performance.

(c) Licencing: awarded licences, certificates, or registrations to show that a job holder has acquired certain skills.

3.4 Proposed SOA-HitLCPS Ontology Model

We view HitLCPS as a combination of humans and machines who interact, communicate, and collaborate to complete their tasks. We use the term machine to refer to any computing system with networking capabilities designed to meet its task cycle autonomously. The machines can be cloud systems and smart devices that are close together in a work environment. To create a self-adaptive human-machine service provision, we need to have a pre-requisite model that includes both human and machine capabilities. To simplify semantic discovery and reasoning, we propose an SOA model for human-in-the-loop CPS, which is expressed as an ontology, called the SOA-HitLCPS ontology model.

Figure 3.2 is a top ontology of our proposed SOA-HitLCPS ontology model, which explains that humans and machines are within an organisation where each node has its function and task, which generally correspond to its context. *Tasks* are roles and activities that have goals to be achieved. In carrying out their roles and duties, each *Physical Thing* may provide services (act as Service Providers) or use services (Service Customers). *Capabilities* are things that enable humans/machines to complete their tasks well. *Context* is the environment, background, setting, or surroundings of events or occurrences of the tasks. Context can be a physical loca-

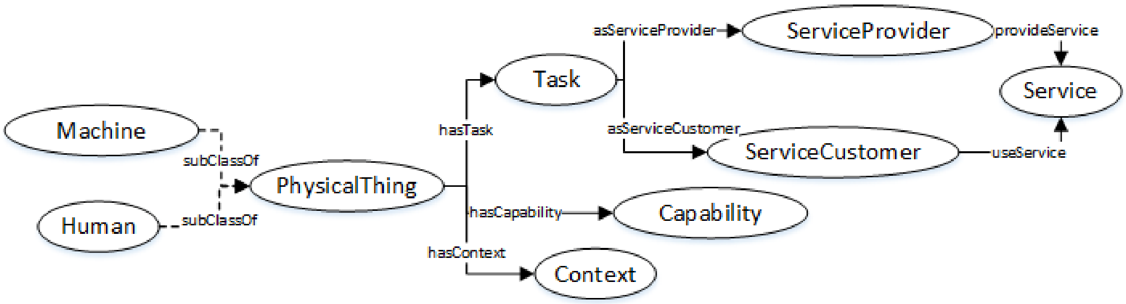


Figure 3.2: Human-machine relationship in HitLCPS

tion, time, temperature, and other contexts in a broader scope related to tasks.

During the process of achieving its goals, the human/machine may need services from others. For example, a bomb disposal technician needs robotic services to cut cables. Or vice versa the robot needs the services of the bomb squad to decide which cable to cut. We can see that each node can be a *Service Provider* or a *Service Customer*.

3.4.1 Service Model

For interoperability reasons, we propose a service model following the OWL-S, which consists of three main parts, namely *Service Profile*, *Process Model*, and *Service Grounding* as shown in Figure 3.3.

The *Service Profile* describes what the service does, and the parameters used, such as input, output, preconditions, effects, service limitations, and non-functional characteristics that distinguish it from other similar services.

The *Process Model* is a specification that explains how the service is used, what constraints must be satisfied and what patterns are required to interact with the service.

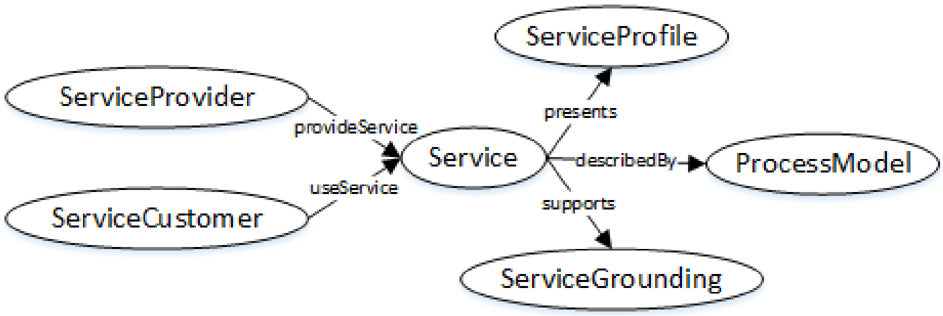


Figure 3.3: Upper layer service ontology

Service Grounding describes how to interact with the service (message format, transport protocol, etc.). In OWL-S, the service grounding is a bridge between syntax- and protocol-oriented WSDL and semantics-oriented OWL.

3.4.2 Service Profile

Service Profile allows providers to advertise their services and also requesters to specify the service capabilities they require. The aim is to support the Service Discovery mechanism to find the most suitable service-customer needs. Each element in Service Profile in Figure 3.4 is described as follows:

- *Service Type* describes the types of service that can be either atomic or composite. Atomic service can be in the form of sensing service, actuating service, or communicating service. Meanwhile, composite service is a combination of several atomic services.
- *Input* refers to the data required by the service to process.
- *Output* is the data produced by the service.
- *Preconditions* are all conditions that must be met (true) before service execution.

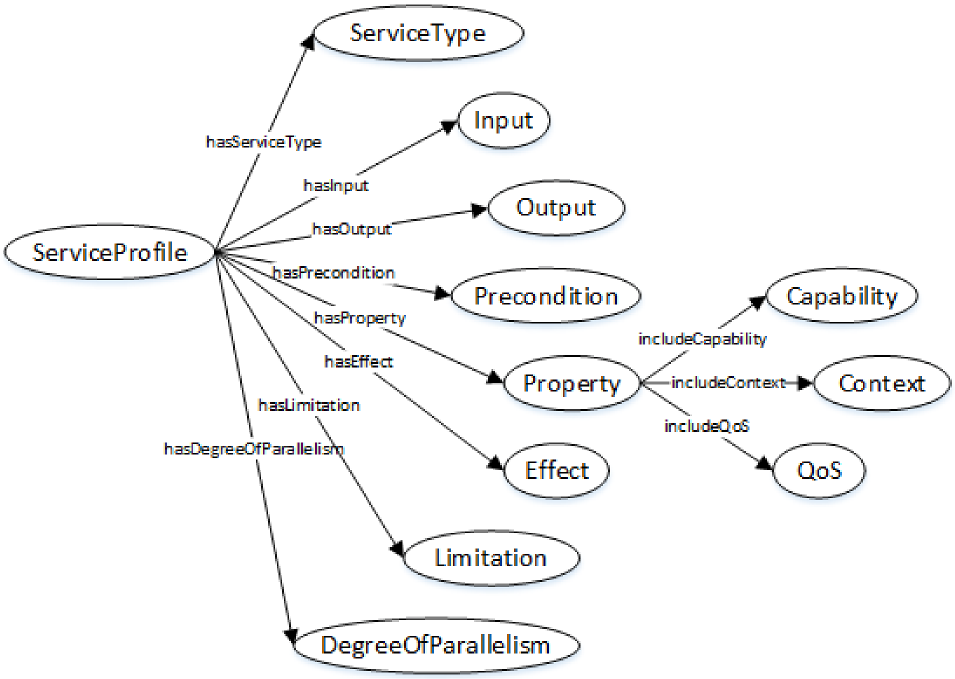


Figure 3.4: Service profile

- *Property* is an attribute that is held by the actor/service provider at that time. These attributes can be related to context, capability model (discussed in the next subsection), and QoS (e.g., reputation, cost, response time, etc.)
- *Effects* are conditions that hold after the service execution.
- *Degree of Parallelism*, adopted from [99], indicates the number of requests this service can serve.
- *Limitations* are things that limit the continuity and availability of services. For example, a service can only be delivered within a specific time frame, a certain distance, a particular location and a particular condition

3.4.3 Human Capability

We argue that human-as-a-service is closely related to occupation because, in essence, humans provide services in every task they do within the scope of their profession.

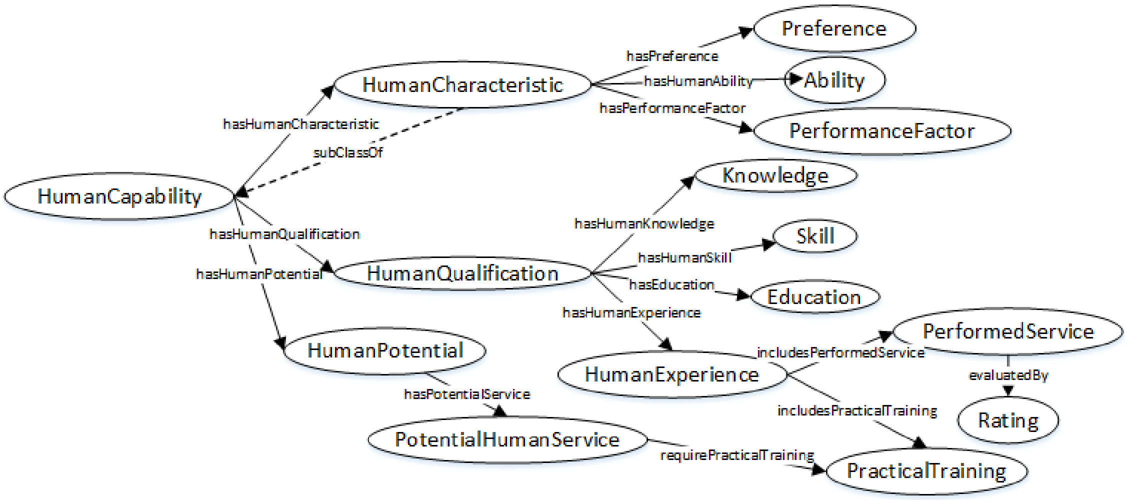


Figure 3.5: Human capability model

An occupation could involve one or more human-as-a-service, atomic, and composite. As human qualification determines the quality of work, we consider it necessary to put knowledge, abilities, and skills as essential components in our proposed human capability model, as shown in Figure 3.5.

- *Characteristic* represents psychophysiological factors [146] that distinguish humans and affect the services provided.

  We express these factors into three categories:

  - *Preferences* correspond to a person's preferences for work environments and outcomes that could affect service availability, such as time, location, and price. Preferences are compatible with O*NET's *occupational interets*.
  - *Abilities* expresses innate human attributes that affect their cognitive, physical, psychomotor, and sensory performance. These ability attributes are usually defined with a measurement scale. Abilities are compatible with O*NET's *abilities*.
  - *Performance factors* are internal and external variables, aspects of hu-

man behaviour and the context (or environment), that can affect human performance reliability. This element is a derivative of the *Work Value* and *Work Style* in the O * NET framework. Scale is used to describe which factors are more dominant than others.

- *Qualification* are attributes that describe a person's appropriateness/fitness, achievement, and quality, which can be either Skill sets, Knowledge, formal Education, or Experience.
  - *Skills* are obtained from training and experience which are defined together with the scale.
  - *Knowledge* refers to domains of expertise or scope/area of work. This pair of qualifications is essential. As an illustration, someone with driving skills and knowledge of city A will find it difficult to drive in city B.
  - *Education* refers to one's level of formal education or degree.
  - *Experience* stands for records of services that have been performed along with ratings obtained from service requesters. The rating system used can vary and may include several assessment criteria. Referring to the O * NET framework, Experience also records practical training (i.e. onsite/on-the-job training).
- *Potential* is defined as the latent human capacity to improve for growth and development [147]. Human skills are developed by knowledge acquired from experience. Not only improving the quality of services in general (improve the skills level), this also opens up new service opportunities (*Potential Service*) that may be provided after new knowledge and skills are acquired. This concept is aligned with the concept of Maximum Human in [148] to maximise the active humans for greater returns in their activity profile, also with Human Capability Theory [149] in which social systems should promote human flourishing.

3.4.4 Machine Capability

Its hardware and software specifications define the computing capability of a machine. Analogously, this is similar to Abilities in humans, but the machine can be upgraded with better component replacement. Skills and knowledge on the machine are the programming logic and datasets provided by the creator. If AI technology is employed, then machines can grow their knowledge (i.e. dataset, ontology) to improve their ability to perform certain functions/services. Machine learning can be done online or offline using shared artefacts or inferred during communication with other nodes. However, to acquire a new type of skill, new logic needs to be inserted into the system. In other words, without reprogramming the machine will not have new services automatically.

3.5 Using our Ontology in Self-adaptive HitLCPS

The definition of our SOA-HitLCPS ontology model is a pre-requisite for supporting future developments for self-adaptive human-machine service provisioning in CPS. Self-adaptivity in HitLCPS can relate to bi-directional cooperation in which machines can help humans or vice versa. Therefore, it is essential to understand machine vs. human behaviour to properly utilise their strengths in a collaborative-oriented environment for optimal results (i.e., not a competition to replace each other).

We instantiated the model using two simple scenarios in the context of a smart healthcare CPS environment. The CPS system connects patients, medical experts, and other smart agents (i.e. machines). We implemented the proposed model as a semantic information model by leveraging OWL standard ontology language and

Protégé [26] editor to evaluate the feasibility of our conceptual model for each scenario. For space limitations, we do not provide instances of all concepts, only those that are essential to demonstrate the feasibility and applicability of our model.

3.5.1 Architecture

Depending on the domain characteristics and requirements, several self-adaptation control-loop patterns can be used [44], be it hierarchical control, master/slave, regional planner, information sharing, or fully decentralised. We show how our ontology model can enrich IBM's MAPE-K reference architecture, where we use the Hierarchical Control pattern. In Figure 3.6, two layers of MAPE-K (Monitor-Analyze-Plan-Execute over a shared Knowledge) feedback loop are implemented using the SOA paradigm. Although it looks like a master/slave pattern, in a hierarchical control pattern, the overall system is controlled by a hierarchical control structure, where each hierarchy level has complete MAPE-K loops.

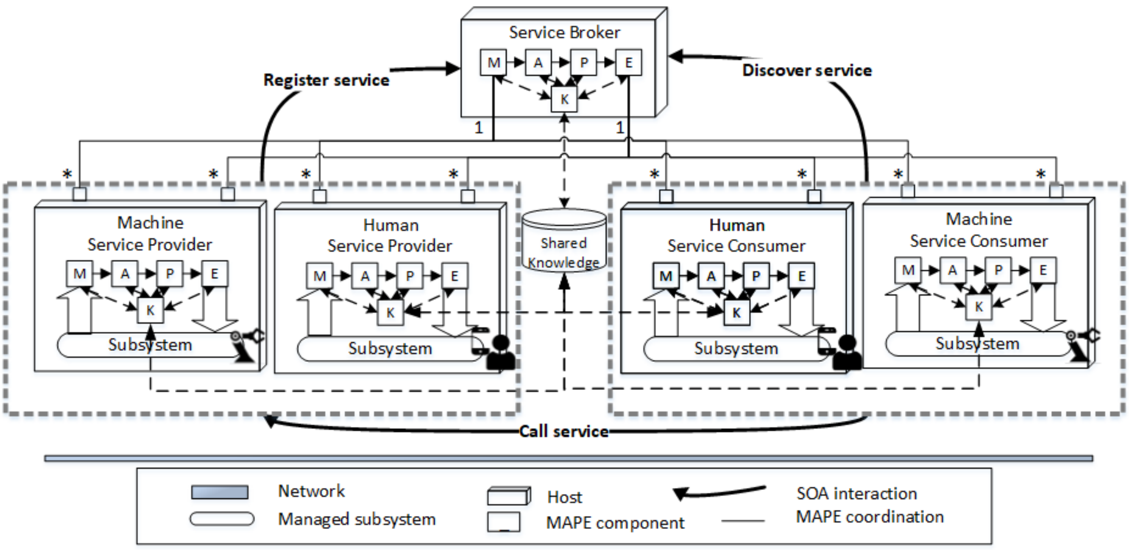


Figure 3.6: Generic SOA model for self-adaptive systems using MAPE-K hierarchical control pattern

MAPE-K loops at different levels interact by exchanging information that contributes to new knowledge stored in a shared knowledge base that other hosts can access.

The top layer is a service broker that carries out service provisioning and service composition, discovers and invokes service implementation candidates that meet the criteria requested and returns the best invocation result or composition plan to the service consumer.

The second layer is self-adaptive systems with MAPE-K loops that interact directly with managed resources or subsystems, representing machines and humans. Depending on the service flow direction, each self-adaptive system at this hierarchical layer can be a Service Provider (when delivering services) and a Service Consumer (when using services).

In service provisioning, the second layer coordinates with the adaptive service broker to achieve optimal results. The hierarchical structure allows the second layer to focus on more concrete adaptation goals, while the higher level can handle adaptation strategies for a broader perspective [150].

3.5.2 Example Scenarios

We present two scenarios in the medical domain to demonstrate the feasibility of our ontology.

Scenario 1

In this scenario, we have a healthcare environment with essential technology to support responsive patient care. The healthcare provider involved is a critical care nurse trained in handling emergency interventions, especially in situations requiring immediate attention to medical device functionality. In this scenario, a machine involves humans to perform beyond its abilities/functionalities.

Andy is a cardiovascular patient who must be continuously monitored using a cardiac output monitoring machine labelled *EcgDev*. *Sisy* is a "Critical Care Nurse" who offers an actuating service *actuatingBySisy* that relies on the sensors and the interface provided by her smartphone to retrieve notifications and context.

This scenario can be instantiated in an ontology illustrated in Figure 3.7 and Figure 3.8.

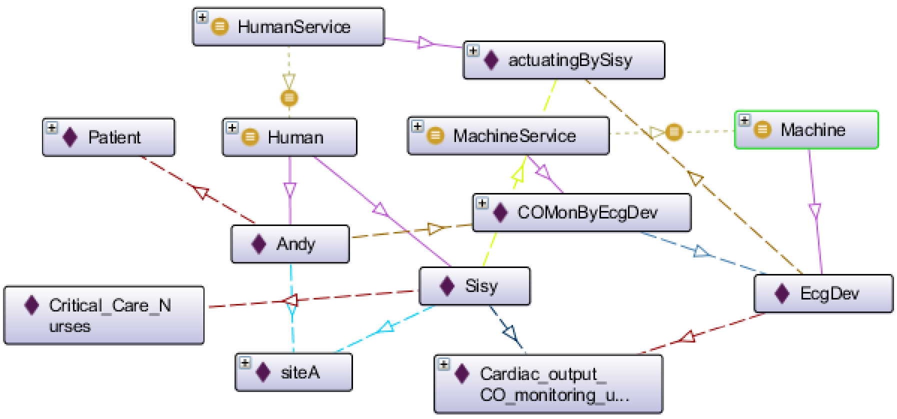


Figure 3.7: Individuals and classes relationships for Scenario 1.

Andy and *Sisy* are instances of the Human class, while *EcgDev* is an instance of the Machine class. *Sisy* possesses a specific capability, represented by a technology skill called "Cardiac Output CO Monitoring Units or Accessories," which enables her to operate *EcgDev*. The operational characteristics (i.e., machine capability) of *EcgDev* are defined by its specifications, such as the ECG sensor and ECG firmware. Additionally, *Andy*, *Sisy*, and *EcgDev* exist within the same context, which is the region of *siteA*.

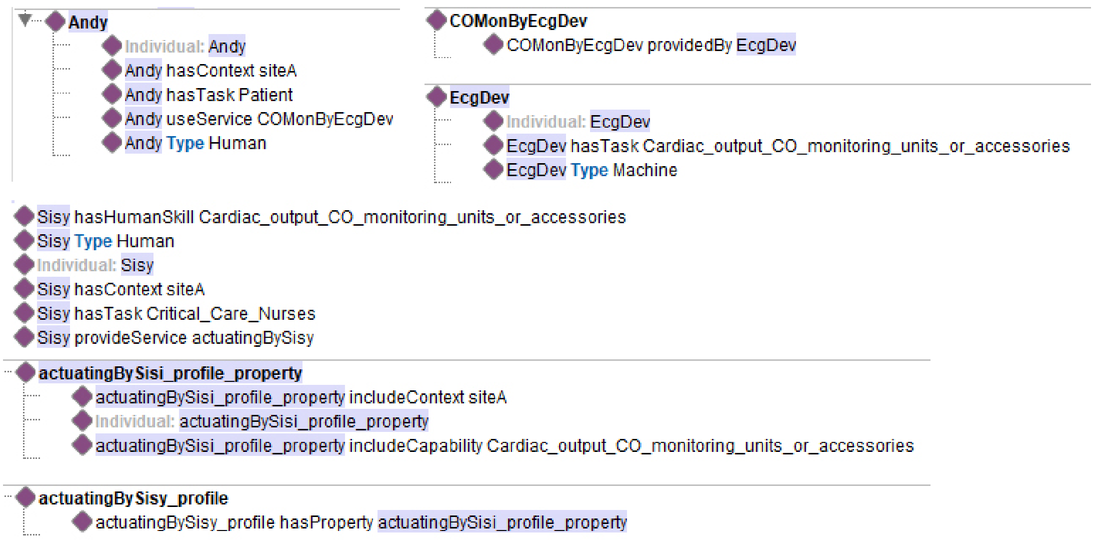


Figure 3.8: Ontology model for Scenario 1.

One day, the cardiac output monitoring machine, *EcgDev*, which was installed on patient *Andy* at *siteA*, loses its signal. In this critical moment, *EcgDev* searches for nearby officers with the skills in "Cardiac Output Monitoring Units or Accessories" to take immediate emergency measures. These measures may include device actuation or emergency care for patient *Andy*. The process is executed by using SPARQL to navigate the Service Profile as follows:

```
SELECT ?service
WHERE {
    ?service soa—HitLCPS:presents ?serviceprofile .
    ?serviceprofile soa—HitLCPS:hasProperty ?property .
    ?property soa—HitLCPS:includeCapability ?capability .
    ?property soa—HitLCPS:includeContext ?context .
    ?capability soa—HitLCPS:hasHumanSkill ?skill .
    FILTER (?context=soa—HitLCPS:siteA  &&
    ?skill=soa—HitLCPS:Cardiac_output_CO_monitoring_units
    _or_accessories)
}
```

The query results indicate that *Sisy* is the most suitable officer to assist *EcgDev*.

Consequently, *EcgDev* submits a service request to *Sisy*.

Upon receiving the notification, *Sisy* promptly rushed to the location of the *EcgDev*. If it turns out to be a false alarm, *Sisy* is responsible for initiating a request for maintenance of *EcgDev* to the Maintenance Dept. After completing the service, *Sisy* receives an assessment by the autonomic manager, a system that evaluates her responsiveness and its impact on the process, verified by her supervisor.

Scenario 2

A health clinic uses web chat as a health service channel. Patients can take advantage of this service to ask questions about clinical services, doctor schedules, book a GP, and get health advice.

In this scenario patient *Adam*, doctor *David*, and chatbot *Cathy* are at layer two that exchange and utilise services from one another. Patients and doctors are represented by smart personal devices that provide an interface (e.g., API) for other entities to interact with their users.

Patient *Adam* accesses the webchat service to consult about the health problems he is experiencing. For every new conversation session request, the *chatbotService* by chatbot *Cathy* is allocated first.

Cathy answers *Adam*'s questions relying on its AI and knowledge from the local and shared knowledge base. During the conversation, *Cathy* acquires information from the chat with *Adam* and stores it on a shared knowledge base for others to use.

Adam repines of discomfort in his head, yet *Cathy*'s answers don't quite satisfy him. With natural language processing and through its MAPE-K loops, *Cathy* detects *Adam*'s emotions and upset. *Cathy* then sends a *service discovery* request to the *Service Broker* with several criteria to maintain customer experience and satisfaction. Based on the conversation, *Cathy* can infer that *Adam* needs a human service with better "Medicine and Dentistry", "Therapy and Counseling" knowledge and "Complex Problem Solving" skills which *Cathy* does not pose.

Based on the criteria given by *Service Broker* discovers and invokes service implementation candidates that meet the invocation criteria using the SPARQL query as follows:

```
SELECT ?service
WHERE {
    ?service soa-HitLCPS:presents ?serviceprofile .
    ?serviceprofile soa-HitLCPS:hasProperty ?property .
    ?property soa-HitLCPS:includeCapability ?capability .
    ?capability soa-HitLCPS:hasHumanSkill ?skill .
    ?capability soa-HitLCPS:hasHumanKnowledge ?knowledge
    FILTER (?skill=soa-HitLCPS:Complex_Problem_Solving
    && ?knowledge IN (soa-HitLCPS:Medicine_and_Dentistry,
    soa-HitLCPS:Therapy_and_Counseling))
}
```

The query above returns the result that the *chatDoctor* service by doctor *David* is a suitable candidate. The Service Broker invokes the *chatDoctor* service for *Cathy*, which then gives *David* access to join the chat session. At this point, *Cathy* becomes a *Service Consumer* (to *David*) as well as a *Service provider* (to *Adam*).

David provides an *adaptation service* for the chatbot *Cathy* to provide a better quality of service in the future. *Cathy* acquires the knowledge from the conversation between patient *Adam* and doctor *David*. This knowledge can be in the form of questions, answers and responses given. However, during the chat session, *Cathy*

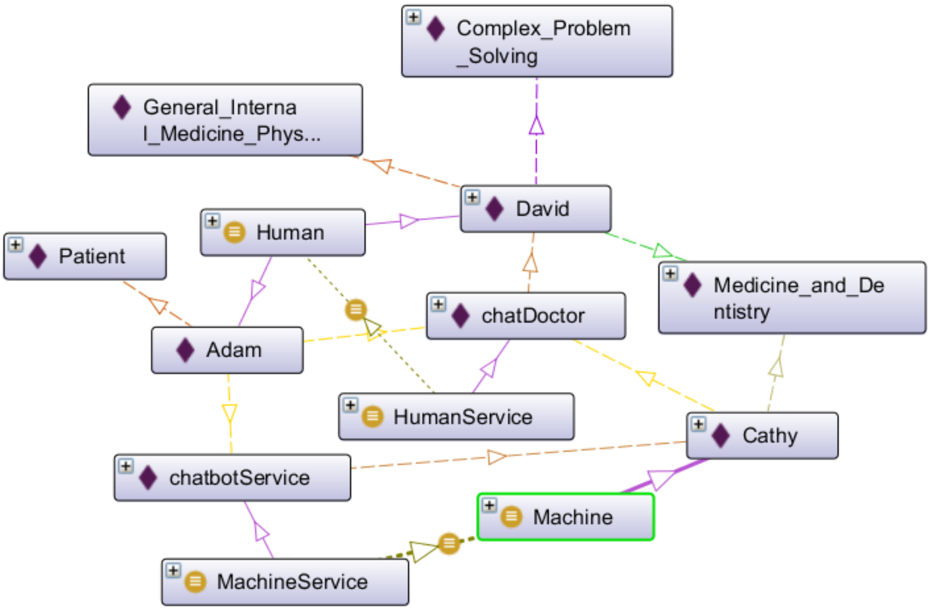


Figure 3.9: Individuals and classes relationships for Scenario 2.

can still provide answer recommendations which can be adjusted by *David*.

David gives some advice to *Adam*, and they agree on a schedule for offline meetings. *Adam*, who was initially upset, ends the session with good satisfaction.

The instantiation of this scenario is shown in Figure 3.9.

By involving chatbot *Cathy*, doctor *David* can get other work done and handle more customers, thus increasing productivity. *Cathy* can learn from *David* to provide more human responses in the future by showing empathy. Meanwhile, *Cathy* can also offer answer suggestions to *David*, especially when it comes to data processing where humans are inferior to AI.

This scenario shows that machines and humans can help each other extend the capabilities of both. Self-adaptive HitLCPS, benefiting from our ontology, can modify its behaviour or structure in response to the changing context (environment, goal, system) that it perceives.

3.6 Evaluation

In the previous section, we have shown how our ontological model can be applied to two different healthcare service scenarios using Protégé [26] editor. Below we provide a qualitative and quantitative evaluation of our ontological model, referring to [151] to ensure our design adheres to certain desirable criteria, such as accuracy, completeness, adaptability, clarity, and consistency. We use HermiT 1.4.3.456 ontology reasoner, which can evaluate whether or not the ontology of input is consistent, define subsumption relationships between classes, and much more. Several test scenarios are employed by adding several instances/individuals and the relations that connect one individual to another. The expected results then be compared with the Inferred Results by HermiT.

Accuracy

Accuracy is a criterion that indicates whether the axioms of an ontology comply with the domain knowledge. We have made every effort to make each element or concept expressed in Class or Relations in this model comply with existing standards and literature. For instance, in our human capability model, we refer to O*NET for a taxonomic approach and integrate it with the Human Capability Theory to add the $HumanPotential$ concept.

To provide accuracy, we ensure that the axioms should constrain an ontology's potential interpretations such that the resulting models are consistent with the users' conceptualisations. In the illustration below, it can be seen that axioms 1 and 2 refer to the facts and conceptualisation of the concept in the given concept definitions 1 and 2.

Given:

Concept 1: Human is a physical thing with human capability

Concept 2: Human service is a service provided by human

Output:

Axioms 1: PhysicalThing and (hasCapability some HumanCapability) SubClassOf Human

Axioms 2: Service and (providedBy some Human) SubClassOf HumanService

Completeness

Completeness measures if the domain of interest is appropriately covered. The domain of interest of this model is to promote human-machine service provisioning so that we ensure our ontology is able to answer several basic competency questions (CQ) related to service delivery. These competency questions are formulised as SPARQL queries towards ontology. We compare our model with the existing related models, HSCD [10] and PE-ontology [96] in Table 3.1. Our model provides the answers to the given CQs, while the other two models require a change in the ontology and develop the concept into several subclasses of ontology (i.e. ontology evolution) to answer questions related to skills, knowledge, and abilities. HSCD does not provide a model for machines/physical entities because it limits their scope to humans, whereas PE-ontology does not take into account humans specifically in their model.

Adaptability

Adaptability measures how far the ontology anticipates its use. It should offer the conceptual foundation for a range of anticipated tasks and allow for methodologies

Table 3.1: Comparison of our SOA-HitLCPS model with other models in answering Competency Questions

Competency Questions	Provision of Answer		
	SOA-HitLCPS	HSCD	PE-ontology
CQ1: Is this human?	✓	✓	R/E
CQ2: Is this machine?	✓	R/E	✓
CQ3: Which human has this ability?	✓	R/E	R/E
CQ4: What services are available?	✓	✓	✓
CQ5: Which service requires this skill?	✓	R/E	R/E
CQ6: Which service requires this knowledge?	✓	R/E	R/E
CQ7: Which services meet the given criteria?	✓	✓	✓
CQ8: Who is the service provider for this service?	✓	✓	✓

✓: instant, R/E: requires evolution

for extension, integration, and adaptation. New tools and unexpected situations should be able to use the ontology. Our proposed ontology enables not only human-machine service provisioning but also machine-only service provisioning and human-only service. However, one can leverage existing concepts in our ontology for other purposes. Our SOA-HitLCPS's human capability is closely related to human resource development functions. Several concepts can be utilised for better provision of training, career planning, promotion, and payroll. Another example is predictive maintenance, which primarily involves foreseeing the system's breakdown to be maintained by detecting early signs of failure to make maintenance work more proactive. Some techniques like oil analysis and vibration analysis (mechanical looseness or weakness) are possible by leveraging the $Machine_Specification$ and $Experience$ concepts on our model.

Clarity

Clarity measures how effectively the ontology communicates the intended meaning of the defined terms. This criterion can be measured by using Class/Relation Ratio (CRR) from [152] that can be formulated as:

$$CRR(O) = \frac{|C(O)|}{|P(O)|} \tag{3.1}$$

where C(O) is the cardinality of the set of classes represented by nodes in O, and P(O) is the cardinality of the set of relations in O.

We compare our SOA-HitLCPS model with HSCD, and PE-ontology in Table 3.2. Each class and relation in the models is assumed as a class and object property in the ontology. Although actually Object Properties, Equivalent Classes, Disjoint Classes, and Subclasses (Subclass of) are counted as relationships, for an apples-to-apples comparison, we only calculate Object Properties and Subclasses to determine P(O). In this illustration, we can see that our model involves more classes and relations than the other two models with the lowest CRR. Lower CRR value means there are more relations/properties to explain a concept (class); provides more clarity.

Table 3.2: Class/Relation Ratio (CRR) Comparison

	SOA-HitLCPS	**HSCD**	**PE-ontology**
C(O)	**46**	**17**	**10**
Object properties	45	10	9
Subclasses	10	6	0
P(O)	**55**	**16**	**9**
CRR(O)	**0.84**	**1.06**	**1.11**

Consistency

Consistency describes that the ontology does not include or allow for any contradictions. We ensure consistency using two different methods. First, we rely on the output of HermiT reasoner [153], which is based on "hyper tableau" calculus that provides efficient reasoning and ontology consistency tests. Moments after the HermiT reasoner is started, HermiT will generate errors if it finds any inconsistencies, and our implementation is free of this. We also ensure there are no inconsistencies by providing no class equivalent to owl: Nothing in the inference results.

Second, we use the Ontoclean [154] methodology to analyse the taxonomy of classes that have subsumption relations (i.e. sub-class, sub-type). Ontoclean has

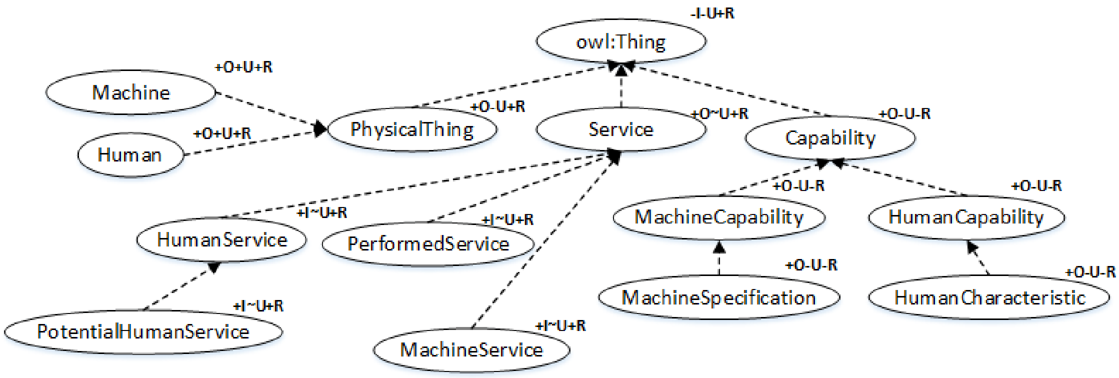


Figure 3.10: The subsumption relationships with their Ontoclean metaproperties

the following rules: given two properties, p and q, when q subsumes p the following constraints apply: if q has anti-rigid (˜**R**) and/or anti-unity (˜**U**) and/or an identity (+**I**) criterion and/or a unity criterion (+**U**) then p must carry the same corresponding criterion/metaproperty. As shown in figure 3.10, the subsumption relationship in our ontology is consistent, according to Ontoclean.

3.7 Threats to Validity

Our ontology may have threats to external validity. In domains that necessitate more specific additional classes, it will require further extension. Conversely, in broader domains that need fewer classes, it should be shrunk and simplified. However, our evaluation results show that our ontology can be useful for human-machine collaboration in a broad sense that views humans and machines as service providers/consumers.

Threats to the internal validity of our ontology relate to the compentency questions (CQs) we use to measure completeness (i.e., comprehensiveness). The CQs we select represent basic competencies that effectively distinguish between machines and humans as service providers, as well as differentiate one service from another.

By employing these CQs, our ontology demonstrates greater completeness than the baseline models. However, we recognise that our ontology may struggle to address CQs in more specific domains, which limits its overall completeness.

3.8 Conclusion

We propose a conceptual SOA ontology model for humans as a service provider in CPS, called the SOA-HitLCPS ontology model. In our model, machines and humans can help each other extend their capabilities; humans can provide sensing, processing, actuating, and promoting adaptation for other nodes within the CPS. Two use case scenarios from the medical domain are used to illustrate how SOA-HitLCPS can be instantiated. As SOA-HitLCPS is an enabler and pre-requisite for engineering a self-adaptive CPS with human-machine collaboration as service providers, we have reported on how self-adaptive reference architecture models such as MAPE-K can be refined and leveraged by SOA-HitLCPS. In addition to establishing the feasibility and applicability of SOA-HitLCPS by means of instantiation on a use case and enrichment of MAPE-K, the evaluation follows standard and commonly used approaches to ontology evaluation, where we evaluate the ontology against criteria that relate to accuracy, completeness, adaptability, clarity, and consistency.

Chapter 4

Fairness-Aware Service Provisioning in HitLCPS: a Heterogenous Crowdsourcing Case

Industry 5.0 utilises the Internet of Things (IoT) and autonomous computing to facilitate human-machine collaboration, where humans and machines coexist in a competitive economic ecosystem. In conventional workplaces, fairness is widely recognised as a driving force behind human motivation, loyalty, and productive collaboration. However, current fairness-aware task allocation methods have primarily focused on homogeneous workers, concentrating on either equity or equality as the sole fairness principle. With the rising trend of diverse worker fleets consisting of autonomous robots/vehicles and humans-in-the-loop as service providers (e.g., crowdsourced logistics), novel approaches are necessary.

This chapter discusses our contribution of a fairness-aware task allocation approach for heterogeneous workers, leveraging digital twins to understand the system's behaviour and facilitate real-time adaptation. Our proposed solution considers equity, equality, and need, utilising the maximum-weight bipartite matching algorithm.

Multiple incentive scenarios are utilised to evaluate the potential of the approach. The experimental results suggest that our multi-objective approach yields better overall fairness in various scenarios than the baselines. This chapter originally appeared as [28].

4.1 Introduction

Industry 5.0 drives increased human-machine interaction in various sectors, including smart logistics [155] that rely on the Internet of Things (IoT) and intelligent infrastructure for on-demand delivery. Companies like Amazon, Postmates, JustEat, and Uber leverage human-machine collaboration to optimise delivery services. However, economic competition between human and machine workers can impact human workers' earnings [156]. Existing fairness-aware task allocation approaches for crowdsourced logistics [109, 108, 110, 107] primarily focus on homogeneous crowdsourcing involving human workers only, assuming relatively equal capabilities. However, these approaches may not be suitable for heterogeneous settings [157] with varying performances among workers.

Unlike machines, humans are driven by a variety of motives [22, 27], including the desire for rewards or incentives. Research in social psychology has shown that fairness is crucial in promoting worker participation, and there is strong evidence that fairness incentives can significantly influence human behaviour [158].

In dynamic crowdsourcing settings, job satisfaction and worker turnover are negatively correlated, highlighting the importance of perceived fairness in fostering worker loyalty. A survey of MTurk workers has shown that fairness is a significant factor in determining job satisfaction and favourable employer behaviour [159].

Intuitively, fairness has been defined in many research in computing as an equal [109, 108], or a proportional distribution of resources [110, 107]. In the Machine Learning community, fairness is related to prediction and is defined in terms of protected attributes and privileged/unprivileged groups to provide equal opportunity [160]. Meanwhile, fairness and justice have been extensively discussed in social psychology literature. We ground our work on theories from social psychology to develop sound foundations for a novel method aimed at attaining fairness in task allocation for human-machine crowdsourced logistics assisted by the digital twin.

Balancing equity, equality, and need in crowdsourced logistics in the physical setup (i.e., live system/production environment) is challenging to observe and adjust the operations. Datasets obtained from the live crowdsourced logistics system can be limited, incomplete, or may not cater to all eventualities, including extreme, stressful, and unanticipated cases at scale and at varying times. Updating strategies/policies in the live system can be disruptive, risky, unsafe, and/or erroneous. Digital twin solutions can address these practical challenges and improve safety in decision-making by enabling organisations to test and validate updates in a risk-free virtual environment before implementing them in live systems. This approach minimises disruptions and prevents unsafe scenarios, as planners and designers can simulate changes, conduct what-if analysis, and refine strategies based on real-time data and/or simulated data. By identifying potential issues within the digital twin, organisations can reduce the risk of errors, allowing for safer and more informed decisions regarding system updates.

This chapter demonstrates how equity, equality, and need as NFRs in software systems are essential in human-machine collaboration. We use a digital twin at design time to figure out how fairness based on equity, equality, and need can be pursued in various incentive scenarios. In addition, we also demonstrate how the digital twin can steer adaptation at runtime to achieve better overall fairness. In a nutshell, this chapter makes the following contributions:

- We present human-machine collaboration in a competitive economy and advocate fairness as an important aspect that one must consider in the task allocation for a diverse fleet of workers, both human-human and human-machine heterogeneous ecosystems.
- We define fairness using three different distributive justice principles (*equity, equality* and *need*) and formulate metrics to measure unfairness on each criterion.
- We implement the fairness principles into the task allocation algorithm for spatial crowdsourcing based on a maximum weight bipartite matching approach to achieve a balanced trade-off between *equity, equality* and *need.* To the best of our knowledge, we are the first to consider *equity, equality*, and *need* explicitly for task allocation in spatial crowdsourcing and human-machine systems.
- We introduce a reference architecture of digital twin for heterogeneous crowdsourced logistics encapsulating the above; the twin can assist the physical system in task allocations involving human-machine collaboration, considering equity, equality, and fairness. We use various incentive scenarios to demonstrate how coining fairness-aware algorithms and the digital twin can assist in design and runtime planning, analysis, and self-adaptation.

4.2 Fairness and Distributive Justice Principles

In social psychology, distributive justice refers to *the* ***perceived fairness*** *of how the burdens and benefits of social cooperation are shared by (distributed among) group* *members*[161]. According to Deutsch [162] and Folger et al. [163], there are three most relevant rules for distributive justice: **(1) equity**, **(2) equality**, **(3) need**.

4.2.1 Equity

The basic units of equity are the inputs or contributions that individuals contribute to a relationship (whether positive or negative) and the outputs that individuals receive from a relationship [164]. The equity principle states that group members should be rewarded in proportion to their contributions or inputs, meaning that those who contribute more should receive greater rewards. Equitable connections between person 1 and person 2 are established when the ratio of output to input is the same for all members, as represented by the formula $O_1/I_1 = O_2/I_2$, where O_i denotes the output of the person i, and I_i represents the input of the person i [165]. Hence, equity will be the dominating basis of distributive justice in cooperative relationships when economic efficiency/productivity is a major aim [162].

Several solutions have been proposed for crowdsourced logistics that satisfy the equity principle, using contributions or inputs such as the number of approved offers [107], completed tasks [108, 107], or distance travelled [108, 110] to determine the total reward or incentive at the end of the day. However, the equity principle is more appropriate for homogeneous systems where the natural capability of workers can be fairly compared. Heterogeneous crowdsourced logistics involve a team of workers with diverse innate capabilities. In this case, the least privileged workers may struggle to earn incentives, as job offers are likely to be allocated to those with higher natural abilities, resulting in fewer opportunities for the least privileged workers. Therefore, in heterogeneous crowdsourced logistics systems, it is imperative to consider other fairness principles.

4.2.2 Equality

The equality rule stipulates that each group member should receive an equal share of the group's outcomes. Equality, or the principle of equal results, functions when there is a sense of group unity and a collaborative environment with the objective of achieving group harmony [164, 162].

Lan et al. [109] aimed for equality by providing workers with the same reward for a uniform number of tasks. However, this approach can result in income inequality if each task carries a different incentive.

Several crowdsourcing companies implement the principle of equality through flat hourly wages [166], where all workers receive the same wage regardless of their productivity and the number of tasks they complete. This principle is generally acceptable in homogeneous systems with relatively uniform worker capabilities and well-distributed tasks.

Flat rate compensation is problematic due to variations in completion times among workers caused by differences in skills and motivation [166]. Without proper control mechanisms, higher-performance workers may be under-compensated, and workers may intentionally lower their performance to receive more gratification. Therefore, the implementation of equality in task allocation must be complemented by other means to ensure its optimisation.

4.2.3 Need

"Those with the greatest needs should be provided with the resources they need to meet those needs" [167]. The need principle of fairness emphasises compensating individuals based on their needs rather than just performance. It prioritises well-being over equity and input and is often considered when collective well-being is the objective [164, 162, 168]. For instance, companies like Google[1] adjust wage levels based on geographic location to account for the cost of living differences.

In human-machine collaboration settings, machines often receive more tasks due to their capabilities, but in competitive economies like crowdsourced delivery, fair compensation should reflect human workers' needs and contributions. Prioritising human workers' needs can create a fair and inclusive environment. However, there is a need for further discussion in the computer science literature on applying the need principle to ensure fair task allocation.

4.3 Motivating Scenario

Consider the imaginary crowdsourced logistics company, LogistiX, which provides an on-demand delivery service that employs a diverse workforce, including cyclists, moped drivers, car drivers, and autonomous vehicles (AVs). The system generally works as follows: the user creates an order list on the vendor's page in the LogistiX client app and makes a payment. The user needs help choosing the type of courier that will deliver his goods. Once the vendor confirms the order, the LogistiX platform will allocate tasks to the appropriate workers. The designated worker will move to the pickup point (e.g., vendor's location, warehouse) to pick up the goods and then bring the goods to the delivery point/drop-off point (i.e., end customer). Workers will then receive incentives for each completed order.

Table 4.1 shows the settings and the assumed characteristics of those groups of

[1] https://economictimes.indiatimes.com/magazines/panache/google-introduces-work-location-tool-to-let-employees-calculate-pay-and-benefits-for-remote-work/articleshow/83768103.cms?from=mdr

Table 4.1: Worker characteristic

Worker type	**Max travel distance (km)**	**Average speed (km/h)**	**Entitlement factor**
Cyclist	3	10	0.20
Moped driver	5	20	0.30
Car driver	10	30	0.40
AV	10	30	0.10

workers for handling deliveries within the city. Using human workers and AVs on gig economy platforms involves a different approach to task allocation and worker availability. Human workers, such as cyclists, moped drivers, and car drivers, can choose their shifts and availability and may work for multiple platforms simultaneously. In contrast, AVs are an in-house workforce that can be deployed by the platform to increase worker availability, particularly during busy times.

However, overreliance on AVs to increase system utility and meet customer demand can have potential downsides. Prioritising AVs over human workers can reduce task availability for humans, leading to reduced earnings and job insecurity. Moreover, overdependence on AVs may lead to technical failures or accidents, endangering the safety of workers and customers and resulting in financial losses for the platform.

Therefore, crowdsourced delivery platforms need to strike a balance between utilising AVs to increase system utility and ensuring that human workers have access to a sufficient number of tasks to maintain their livelihoods. This can involve implementing policies and incentives that encourage the use of AVs responsibly and sustainably while providing support and opportunities for human workers to continue participating and thriving in the platform ecosystem.

There are three incentive scenarios commonly used by crowdsourced delivery platforms, which are:

Scenario 1 (IS1). Per item incentive: In this scenario, the reward given to workers is not proportional to the distance of delivery but rather linearly dependent on other factors, such as the price of the items. Hence, the more expensive the goods are, the higher the reward the workers will receive upon successful delivery.

Scenario 2 (IS2). Hourly payment (Equality-based): Workers receive a fixed hourly wage regardless of the number of orders and distance travelled, except during peak hours when they earn an additional fixed bonus for each completed order. To increase their earnings, workers must work more shifts.

Scenario 3 (IS3). Per-mile incentive (Equity-based): In this scenario, the workers are given a per-mile incentive along with incentives for pick-up and drop-off. They are guaranteed a minimum reward for every task assigned to them. The reward structure employed in IS3 is based on the delivery distance, meaning that the greater the distance, the higher the earnings of the workers.

In addition to the three incentive schemes above, other incentive schemes with gamification, surge pricing, and performance-based bonuses can be used as additional incentives to increase workers' loyalty, participation, and satisfaction under certain conditions. We use the three scenarios above to reduce complexity so we can focus on ensuring fairness and avoid potential downsides associated with additional incentives [169, 170].

As previously mentioned in Section 4.2, every reward scheme scenario presents potential fairness challenges without a task allocation mechanism that adheres to other fairness principles.

To maintain its service level, the LogistiX platform aims to implement a fairness-aware task allocation system that promotes competition while prioritising inclusivity and harmony.

4.4 Problem Definition

Developing a fair strategy requires considering stakeholders' values and preferences to meet specific fairness objectives [162]. Fairness perception is subjective, with individuals prioritising either equality or equity. However, in some cases, it may be necessary to consider the principle of need alongside equity and equality [168].

We develop a task allocation strategy in crowdsourced smart logistics environments involving human-machine collaboration while balancing equality, equity and need, inspired by theories from social psychology. The objective is to minimise inequality, inequity, and need-unfairness among workers while still maintaining the system's utility to allocate as many tasks as possible. We leverage digital twin as a medium for what-if analysis and continuous evaluation for fairness-aware task assignments.

In the following, we specify the requirements, formulate the problem and models, provide definitions, and describe assumptions:

4.4.1 Fairness requirements

The crowdsourced logistics system in our motivating scenario aims to allocate tasks and rewards to the workers as fairly as possible by accommodating the three distributive justice concepts in Section 4.2. Therefore, we divide the fairness requirement into three sub-requirements, as follows:

R1. (Equity) Workers shall receive a reward as proportional as possible to their travel distance. Meaning that the greater the distance travelled, the greater the wages received. This principle is consistent with the incentive scenario IS3 but is not guaranteed in the other scenarios. Even in scenario IS2, workers who do nothing earn as much as those who work more.

R2. (Equality) Workers shall receive income as equal as possible. Workers should receive the same total income regardless of their distance. However, due to the variety and dynamism of the locations of tasks and workers, this principle is only guaranteed to be satisfied in the incentive scenario IS2. In other incentive scenarios, it will be challenging to realise.

R3. (Need) Workers shall earn as proportional as possible to their entitlement factor. As in Table 4.1, we provide different entitlement factors for each worker group based on their operating expense to represent their need. Our approach will prioritise workers with higher entitlement factors. Therefore, we give AVs the lowest entitlement factor value so that human workers receive higher priority in task allocation.

4.4.2 Preliminaries

Preliminary definitions and assumptions used in this study are provided below. We summarise frequently used symbols in this chapter, particularly in Section 4.4.2 and Algorithm 1, in Table 4.2.

Definition 1. Workers. w_i represents the i^{th} worker, and W represents the set of workers. The worker is a tuple of $(h, l, v, m, q, n, e, p, r, o)$. Here, h denotes w_i's shift period, specifying the start and end of the shift, l represents the coordinate of w_i's current location, v denotes travelling speed, m is the maximum travel distance of w_i, q is the total distance travelled by w_i, n denotes entitlement factor, e represents total earning, p is the total profit of w_i, and r denotes the net-payoff ratio. w_i

Table 4.2: Description of symbols.

Symbol	Description
w_i	the i$^{\text{th}}$ worker
$w_i.h$	shift time period (begin b, end e) of w_i
$w_i.l$	current location (coordinate x, coordinate y) of w_i
$w_i.v$	travelling speed of w_i
$w_i.m$	maximum travel distance of w_i
$w_i.q$	total distance traveled by w_i
$w_i.n$	w_i entitlement factor
$w_i.e$	w_i's total earning
$w_i.p$	w_i' total profit
$w_i.y$	w_i's payoff ratio
$w_i.r$	w_i's net-payoff ratio
t_j	the j$^{\text{th}}$ task
$t_j.y$	t_j's time period (begin b, end e)
$t_j.u$	t_j's pickup location
$t_j.g$	delivery location of t_j
$t_j.f$	task reward of t_j
W	set of w_i
T	set of t_j
W_k	set of w_i in group k, $W_k \subseteq W$
d_{ij}	travel distance for worker w_i to finish t_j
h_{ij}	completion time of task t_j by worker w_i
p_{ij}	potential profit for the worker w_i from completing task t_j
PR_{ij}	potential profit per distance unit for the worker w_i from completing task t_j
R_i	priority ranking value of w_i according to his/her position in W'
V_{ij}	ratio of potential profit p_{ij} to $w_i.p$
G	worker-task bipartite graph
u_{ij}	weight of edge (w_i, t_j)
M	matching graph

can only be assigned to a task where its pickup and drop-off locations are within $w_i.m$, and w_i must be able to finish the task before the task expires and before his/her shift ends. At a time, w_i can only receive one task assignment. She/he must finish the assignment before being given the next one. Once a task is finished, the worker earns a reward according to the incentive scenario, and then the values of $w_i.q, w_i.e, w_i.p, w_i.r, w_i.y$ are updated. After completing the task, we assume the worker will remain at the drop-off location until the next assignment is received.

Definition 2.Tasks. Tasks take the form of spatiotemporal deliveries: workers must move to the pickup location to collect the item and then continue to a drop-off location to deliver the item to the receiver within a certain period. We represent a set of all tasks with T and the j^{th} task with t_j. Each task is a tuple of (y, u, g, f). Where y represents the time period, pickup location is denoted by u, delivery location is denoted by g, and f denotes task reward.

Definition 3. Basic entitlement. The minimum compensation needed to satisfy a worker's needs, denoted as c_{ij}, is referred to as the *basic entitlement* and is formulated in Equation 4.1. Any reward exceeding the basic entitlement is considered profit (see Equation 4.2.

$$c_{ij} = w_i.n * d_{ij} \tag{4.1}$$

where $w_i.n$ is the entitlement factor of w_i (see Table 4.1, and d_{ij} represents the travel distance for worker w_i to finish t_j.

We propose the concepts of *basic entitlement* and *profit* to anticipate incentive scenarios that are disproportionate to the entitlement factor (i.e., disregard the type of workers), such as scenario IS3, and are disproportionate to the distance as well, such as scenarios IS1 and IS2. Also, to ensure that workers are assigned to tasks whose rewards satisfy their basic entitlement.

Definition 4. Profit. The profit of w_i from executing task t_j, represented as p_{ij} in Equation 4.2, is the remaining reward after deducting the basic entitlement:

$$p_{ij} = t_j.f - c_{ij} \tag{4.2}$$

$t_j.f$ represents the reward of doing task t_j, and c_{ij} denotes worker w_i's basic entitlement of doing task t_j.

Definition 5. Payoff ratio. $w_i.y$ denotes the worker w_i's payoff ratio formulated in Equation 4.3 as the ratio of total earnings to the total distance travelled.

$$w_i.y = \frac{w_i.e}{w_i.q} \tag{4.3}$$

Definition 6. Net-payoff ratio. The worker w_i net-payoff ratio, denoted as $w_i.r$ in Equation 4.4, is the ratio of total profit $w_i.p$ to the total distance travelled $w_i.q$. After completing the task, the value of each worker's net-payoff ratio $w_i.r$ will be updated.

$$w_i.r = \frac{w_i.p}{w_i.q} \tag{4.4}$$

4.4.3 Unfairness Indices

Definition 7. Inequity. Inequity can be defined as the dispersion of the ratio of output (i.e., total earnings) to input (i.e., total distance travelled) among workers. Let Y represent the set of workers' payoff ratio $w_i.y, w \in W$. Therefore, inequity can be measured by the coefficient of variation of Y, denoted as $CV(Y)$ in Equation 4.5, adopted from [107] to measure unfairness.

$$CV(Y) = \frac{\sigma(Y)}{\mu(Y)} \tag{4.5}$$

where $\sigma(Y)$ and $\mu(Y)$ are the standard deviation and mean of Y, respectively.

Definition 8. Inequality. The metric commonly used to measure inequality is the Gini coefficient [171], which determines how much an income distribution deviates from being absolutely equal. The Gini coefficient can be calculated as half of the relative mean absolute difference, which is mathematically equivalent to the Lorenz curve definition. Given ξ as a set of workers earning $w_i.e$, s.t. $w \in W, w_i.e \in \xi$, the

Gini coefficient of ξ, denoted as $GN(\xi)$, can be formulated as Equation 4.6.

$$GN(\xi) = 0.5 \cdot \frac{\Delta(\xi)}{\mu(\xi)} \tag{4.6}$$

Where $\Delta(\xi)$ denotes the mean absolute difference of ξ, and $\mu(\xi)$ is the mean of ξ.

Definition 9. Need Unfairness. Adopting Jain's fairness index [172], need unfairness (NF) can be formulated as Equation 4.7.

$$NF = 1 - \frac{(\sum_{k=1}^{z} X_k)^2}{z \cdot \sum_{k=1}^{z} X_k^2} \tag{4.7}$$

where z represents the number of groups of workers (e.g., we have $z = 4$ in our motivating scenario). X_k represents the relative allocation, defined as $X_k = A_k/S_k$ where A_k denotes the actual share of rewards allocation for the group of workers $W_k, W_k \subseteq W$ and S_k represents the optimal/ideal share for W_k. A_k on Equation 4.8 is defined as the sum of the reward of all workers in the group W_k divided by the sum of the reward of all workers in W:

$$A_k = \frac{\sum_{w_i \in W_k} w_i.e}{\sum_{w_i \in W} w_i.e} \tag{4.8}$$

and S_k is given by Equation 4.9.

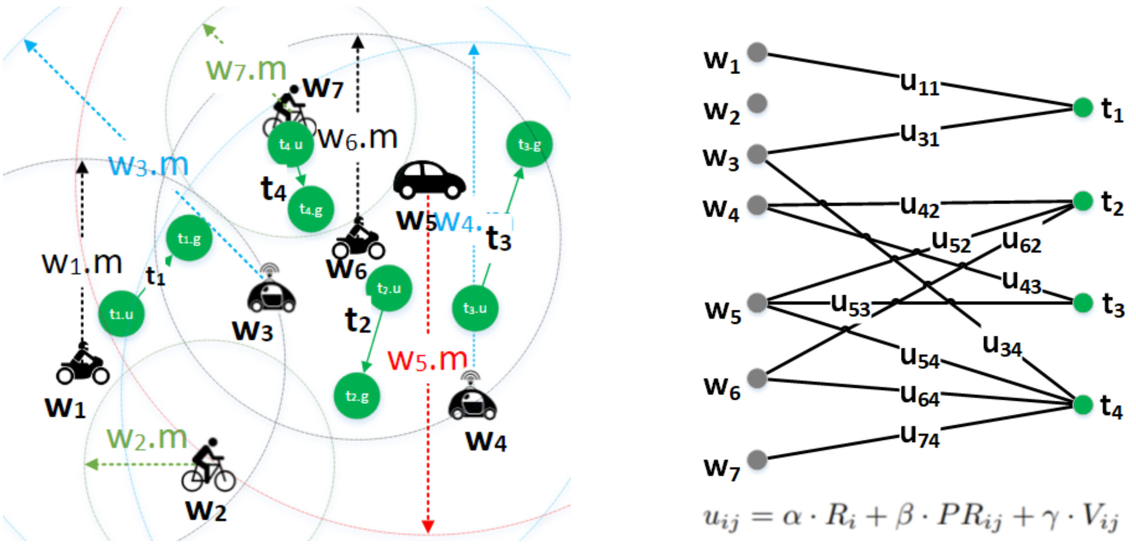
$$S_k = \frac{W_k.n}{\sum_{k=1}^{z} W_k.n} \tag{4.9}$$

$W_k.n$ denotes the entitlement factor of $w_i \in W_k$ (e.g., $W_k.n$ is equal to 0.20 for the group of cyclists)

4.4.4 Problem formulation

Therefore, the fairness-aware task assignment problem can be formulated as a multi-objective optimisation problem with spatiotemporal constraints, as follows:

$$\begin{aligned} \min \quad & (CV(Y), GN(\xi), NF) \\ \text{s.t.} \quad & w_i.m \geq d_{ij} > 0 \\ & h_{ij} \leq t_j.y(e) \\ & h_{ij} \leq w_i.h(e) \end{aligned} \tag{4.10}$$



(a) (b)

Figure 4.1: (a) Each task t_j requires worker w_i to move to the pickup location and then to the dropoff location. Every worker is only available to perform tasks within their maximum travel distance $w_i.m$. (b) The bipartite graph matching $G = (W, T, E)$

4.5 Fairness-Aware Online Task Allocation

We represent workers and tasks as a bipartite graph G, as shown in Figure 4.1(b). The graph $G = (W, T, E)$ is given at each assignment batch, where the worker set W

Algorithm 1: Fairness-Aware Maximum Weight Matching (FMWM)

```
Input   : Set of unassigned tasks T = {t_1, t_2, ..., t_j}, Set of idle workers W = {w_1, w_2, ..., w_i}, α, β, γ
Output  : A tuple set of assignment Ω = {(w_1, t_3), (w_3, t_14), ..., (w_i, t_j)}
W' ← sorted(W, key= Q)                                  // Q is different for each scenario
Initialise graph G
for each w_i in W' do
    G ← addNode(w_i, bipartite=0)
end
for each t_i in T do
    G ← addNode(t_j, bipartite=1)
end
i ← 0
for each w_i in W' do
    for each t_j in T do
        d_ij ← calculateDistance(w_i.l, t_j.u) + calculateDistance(t_j.u, t_j.g)
        if (d_ij ≤ w_i.m) and passTimeCheck(w_i, t_j) then
            c_ij ← w_i.n * d_ij                         // cost to perform t_j
            p_ij ← t_j.f − c_ij                         // potential profit from completing t_j
            PR_ij ← p_ij / d_ij                         // potential profit per distance unit
            R_i ← (len(W')-i)/len(W')                   // priority rank value
            if w_i.p > 0 then
                V_ij ← p_ij / w_i.p // ratio of potential profit from t_j to total profit
            end
            else
                V_ij = 1
            end
            normalise(PR_ij, V_ij)
            u_ij ← α * R_i + β * PR_ij + γ * V_ij
            G ← addEdge(w_i, t_j, u=u_ij)
        end
    end
    i ← i + 1
end
M ← maxWeightMatching(G, maxcardinality=True, weight=u)
for each e in M do
    Insert tuple e(w_i, t_j) into Ω
end
return Ω
```

and the task request set T are the bipartite nodes at each side, and every worker-and-task pair $e(w_i, t_j) \in E$ has a utility u_{ij} weight function $u : E \rightarrow \mathbb{Q} > 0$. Therefore, G has a number of vertices, given by $a = |W| + |T|$, and b number of edges, given by $b = |E|$. We approach this assignment problem as a maximum weight bipartite matching (MWM), also used in [173], as we want to find a matching M such that the weight of matching M, given by $u(M) = \sum_{e \epsilon M} u(e)$, is maximized among all matchings. We describe an **Algorithm 1**, called **Fairness-aware Maximum Weight Matching (FMWM)**, which works as follows:

Initially, the available workers $w \in W$ are sorted in ascending order based on Q, resulting in W'. For scenarios IS1 and IS3, we sort W in ascending order using the keys $Q = (w_i.r, -w_i.n, w_i.p)$, which prioritise task allocation to workers with lower net-payoff ratios, higher entitlement factors, and lower total profit. The order of the keys indicates their priority in sorting, and the negative sign (-) means that the key is sorted in the opposite direction. In the IS1 scenario, workers with lower net-payoff ratios will receive a higher rank. In the case of employees with the same net-payoff ratio, the ranking will be determined based on the entitlement factor using reverse ordering. A higher entitlement factor will result in a higher ranking. If workers have the same net-payoff ratio and entitlement factor, they will be ranked based on total profit, with a higher ranking given to those with lower profit. In scenario IS2, employees with greater profits receive fewer tasks. Therefore, for IS2, we reverse the keys' order to $Q = (w_i.p, -w_i.n, w_i.r)$ and sort W in descending order to minimise inequity.

Next, we initialise the bipartite graph G. Then, in the bipartite graph G, we add each available worker w_i in W' and the unassigned task tj in T as nodes. Then we calculate d_{ij}, the total distance travelled by the worker to accomplish each task t_j in T, for each worker w_i in W'. If w_i fits the spatiotemporal conditions to accomplish task t_j (Alg. 1 line 13), we calculate the worker's basic entitlement to perform task c_{ij} (Equation 4.1), task's potential profit p_{ij} (Alg. 1 line 15), task's potential profit per distance unit PR_{ij} (Alg. 1 line 16), the ratio of each task's net profit to the total profit earned by the worker V_{ij} (Alg. 1 line 18-23), and priority ranking value of w_i in G denoted as R_i (Alg. 1 line 17).

Then we normalise PR_{ij} and V_{ij} using min-max feature scaling to make each of them within the same range $[0, 1]$. After that, we add an edge (w_i, t_j) into G with weight u_{ij}. The weight u_{ij} is a given by $u_{ij} = \alpha \cdot R_i + \beta \cdot PR_{ij} + \gamma \cdot V_{ij}$, where the default values of α, β, and γ are 0.33, respectively. However, the values of α, β, and γ will be dynamically adjusted during runtime adaptation to achieve improved overall fairness. This approach prioritises workers according to their entitlement factor and net-payoff ratios to receive more profitable assignments. Tasks are assigned fairly based on workers' capabilities, and income differences are minimised.

After G is complete, any maximum weight matching algorithm (e.g., Hungarian, Blossom algorithm, or any other algorithms discussed in [174]) can be used to generate M. Each edge of e is turned into a tuple (w_i, t_j) and added to the assignment set Ω when M is obtained.

The quicksort algorithm used on line 1 has a time complexity of $\mathrm{O}(|W|\log|W|)$, and the nested loops in lines 10-30 have a time complexity of $\mathrm{O}(|W||T|)$. The maximum weight matching algorithms on line 31 have running times ranging from $\mathrm{O}(|W|\log|W| + |W||T| + b\sqrt{a}\log N)$ to $\mathrm{O}(\mathrm{poly}(a))$ at the upper bound, where N is the maximum value of u_{ij}, a is the number of vertices given by $a = |W| + |T|$, and b is the number of edges given by $b = |E|$, depending on the maximum weight matching (MWM) algorithm used in line 31 of our algorithm [174].

4.6 Digital Twin Architecture of Heterogenous Crowdsourced Logistics

We propose a reference architecture for digital twins in heterogeneous crowdsourced logistics, consisting of three layers: the digital twin layer, the asset layer, and the decision layer (see Figure 4.2). Our digital twin uses a hybrid of event-driven simulations and virtual representations to exhibit Cyber-Physical interactions between physical assets (crowdworkers and customers) and Cyber-Cyber relationships [78] with the crowdsourcing platform. The digital twin can be deployed as a single integrated service or multiple microservices, communicating with assets through various protocols and messaging patterns via API. This architecture promotes flexibility and independent deployability while providing essential components for heterogeneous crowdsourced logistics.

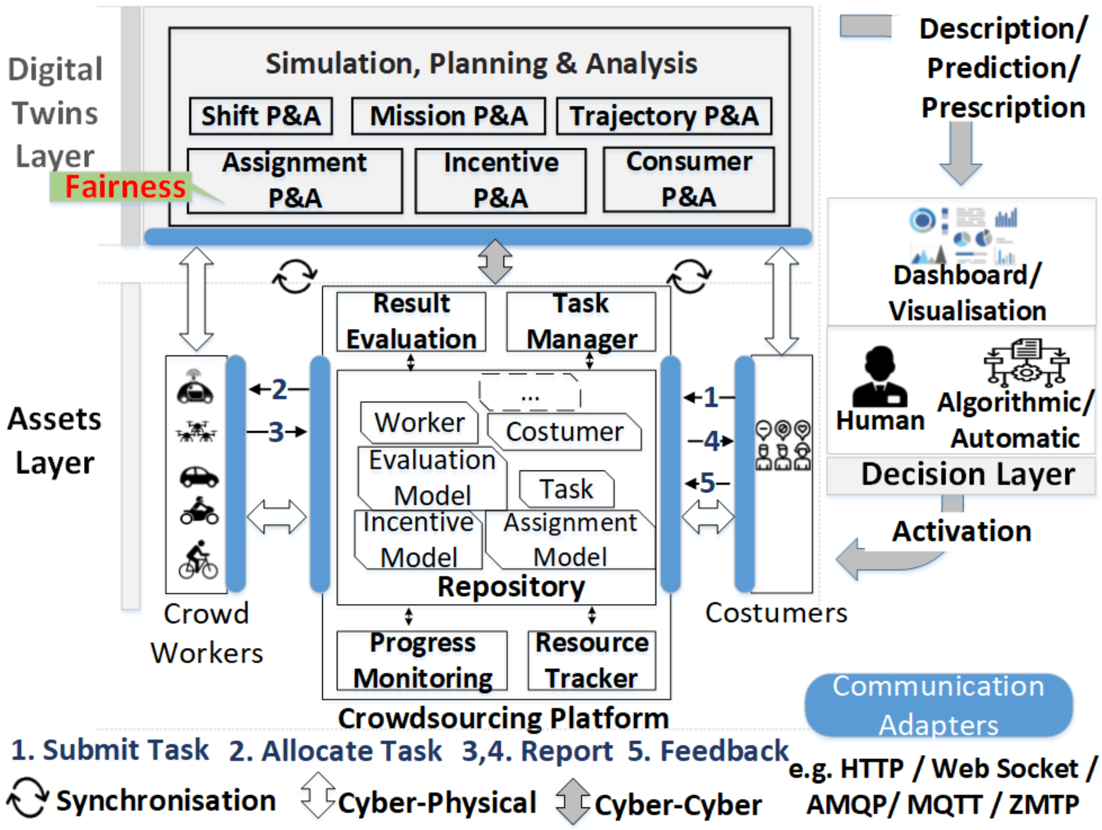


Figure 4.2: Digital twin architecture for heterogeneous spatial crowdsourcing.

4.6.1 Digital Twin Layer

A crowdsourced logistics platform like LogistiX relies on workers and customers to generate revenue. Using various descriptive, diagnostic, prognostic, and prescriptive studies, the digital twin can assist crowdsourcing platforms in developing employee and customer retention strategies. The digital twin for heterogeneous spatial crowdsourcing can incorporate the subsequent analysis and planning elements, utilising the Internet of Things (IoT), Artificial Intelligence (AI), Cloud Computing, and Big Data [155] in the following planning & analysis (P&A) components:

- *Shift P&A*: This is a critical component for worker scheduling, determining worker count per shift, and anticipating task demands because shifts have distinct order types and location characteristics.
- *Mission P&A*: This component uses historical data to determine the best worker location at shift start or after task completion [175]. It communicates with the *Resource Tracker* asset for worker instructions.
- *Trajectory P&A*: Utilises *Progress Monitoring* data to help workers find optimal routes, anticipate risks, and detect anomalies for worker safety.
- *Assignment P&A*: Identifies optimal assignment strategies based on historical data [176], considering fairness, payout ratio, and worker location.
- *Incentive P&A*: This component develops economic and social incentives to foster worker loyalty. The output of the Incentive P&A is the incentive and evaluation or rating model used by the *Result Evaluation* asset.
- *Product P&A*: Implements a recommendation system for sellers and buyers to increase sales, including upselling strategies and tailored product offerings.

4.6.2 Asset Layer

In heterogeneous spatial crowdsourcing, the asset layer may involve devices with diverse computing functions and capabilities. Human workers, autonomous workers, and consumers/task-requesters are all *passive assets*, relying on digital twin intelligence to increase productivity. The crowdsourcing platform is the *active asset*, with sufficient computational capacity to monitor, plan, and analyse but requiring the digital twin for advanced analysis and investigating complex scenarios. The platform has several main components, including Progress Monitoring, Resource

Tracker, Task Manager, and Result Evaluation, which work together to assign tasks, track progress, and provide rewards/incentives.

4.6.3 Decision Layer

In the *decision layer*, the digital twin offers decision support services to humans, enabling them to make judgements in crucial situations using information and visualisations accessible through the dashboard. This layer could be implemented as the View component in the Model-View-Controller (MVC) software architecture pattern to communicate with human users, while the digital twin contains the Model and Controller components. Alternatively, if all decision-making is performed algorithmically, this layer may merge with the digital twin layer.

We demonstrate how our reference architecture for digital twins may benefit decision-makers at design time in Section 4.7.3 and how it can help govern self-adaptation at runtime in Section 4.7.4.

4.7 Evaluation and Discussion

The following research questions (RQs) serve as the basis for our evaluation using two different instantiations of our proposed digital twin architecture:
RQ4.1: To what extent can digital twin analysis and simulation guide analysts in achieving better fairness for the chosen incentive scenarios?
RQ4.2: How can the analysis assist in a better understanding of the impact of batch size on unfairness indices (in Section 4.4.3) and the number of unassigned tasks?
RQ4.3: How can runtime adaptation, supported by the digital twin, better satisfy

the fairness requirement and promote social welfare?

4.7.1 Experiment Setup

Digital twins are used to simulate and analyse two types of synthetic datasets with different settings, as shown in Table 4.3. Each type A dataset uses a constant batch size for each assignment. In contrast, the type B datasets use a dynamic batch size, representing real-world situations. We generate tasks and workers randomly in a 500x500 Euclidean grid space, representing an area of 25 km^2. We have 100 workers per shift with no overlapping time between shifts. Workers are assumed to only change locations upon receiving a new assignment and remain at the drop-off location after completing the task until the next assignment. All tasks in the same batch have the same start and expiration times. The workers move at a constant speed (according to the type of worker) with a Euclidean trajectory distance.

Table 4.3: Synthetic dataset generation settings and parameters

Datasets type	Type A	Type B
Incentive scenarios	IS1, IS2, IS3	IS3
Number of shift	4	4
Assignment per shift	5	5
Workers per shift	100	40
Tasks batch size	25,50,80,100,200	random[10,40]
Number of assignments	20 (5x4 shifts)	20 (5x4 shifts)
Task reward	{IS1=random[200,1000]}, {IS2=2000}, {IS3=1.5 per distance unit + 140 (pickup)+ 110 (drop-off), min. reward 350}	{IS3=1.5 per distance unit + 140 (pickup)+ 110 (drop-off), min. reward 350}
Task expiry period	60 min	60 min
Worker shift period	120 min	120 min
Distribution	random dist.	random dist.

Table 4.4: The comparison of existing fairness-aware task-allocation approaches in spatial crowdsourcing.

App-roaches	**Assign-ment**	**Solution(s)**	**Fairness objective(s)**	**Type of workers**	**Adaptation Strategy**
FATP [109]	SAT	Heuristics	Equal number of task allocations. *(Equality)*	Single	Rule-based
F-Aware [107]	WST	Bipartite graph	Proportionate earnings from the assigned task to the total rewards of the accepted offers. *(Equity)*	Single	N/A
FETA [108]	SAT	Bipartite graph	Equal distribution of schedules and rewards. *(Equity)*	Single	N/A
FGT & IEGT [110]	SAT	Heuristics, Game theory	Equal ratio of workers' earnings to workers' travel time. *(Equity)*	Single	Heuristics
FMWM & Adaptive FMWM (Our proposal)	SAT	Bipartite graph & Heuristics	A balance between (1) an equal amount of earning, (2) an equal ratio of workers' earnings to distance travelled, (3) proportionate earning to the entitlement factor. *(Equity, Equality, Need)*	Multiple	Heuristics, Evolutionary

* SAT (Server Assigned Tasks), WST (Worker Selected Tasks)

4.7.2 Comparative Approaches

Exact algorithms (e.g., maximum cardinality bipartite matching) and greedy-based approximation algorithms are commonly used to solve the static matching problem to maximise the number of assigned tasks [177]. Table 4.4 highlights that the fairness-aware techniques currently available for task allocation in spatial crowdsourcing are designed with varying task assignment contexts and requirements. Furthermore, they adopt different fairness objectives, definitions, and metrics, with a sole focus on homogeneous workers (i.e., a single type of human worker). Consequently, employing these methods as a performance baseline may not be appropriate.

Alternatively, we compare FMWM (algorithm 1) with the following approaches to demonstrate how one may attain fairness by compromising equity, equality, and need in each scenario by instantiating our digital twin architecture.

Basic Greedy (BG) [178]

The main idea of Greedy is to assign every incoming task to the closest available service provider that has not yet been assigned [178].

Fairness-Aware Greedy (FG)

This approach is adopted from *FW-Greedy* as described in [108] as "always greedily assign the task to the most unfair worker". The goal is to assign a job with the highest possible payoff ratio and the shortest distance for the worker with the lowest net-payoff ratio and profit. Initially, available workers $w \in W$ are sorted in ascending order based on their current net-payoff ratio $w_i.r$ and profit $w_i.p$ resulting in W'. For each $w_i \in W'$, sort unassigned tasks set T by their potential profit per distance unit in descending order and distance in ascending order, resulting in T'. While there is an unassigned task in T', take the first element t_0 in T' and check if w_i fits the spatiotemporal conditions to accomplish task t_0. If this is the case, add (w_i, t_0) to the Ω assignment set. Otherwise, set t_0 for the next t in T' and repeat the check. This process is repeated until there are no more tasks in T.

Nearest Neighbour Priority - Maximum Weight Matching (MWM)

To address the spatial aspect of the problem, existing works in spatial crowdsourcing task allocation mainly use the Nearest Neighbour Priority (NNP) approach [179, 107]. We implement NNP using *max_weight_matching* function by [180], which is based on the "blossom" method for locating augmenting paths and the "primal-dual" method for finding a matching of maximum weight. The flow of this algorithm is similar to FMWM. The difference is that this algorithm does not sort workers. It uses $1/d_{ij}$ as the weight u_{ij} for every worker-and-task pair $e(w_i, t_j) \in E$. We attempt to assign as many tasks as possible while minimising the distance as little as possible and satisfying the spatiotemporal constraints.

Adaptive Fairness-Aware Maximum Weight Matching (Adaptive FMWM)

The default values of α, β, and γ used by FMWM are 0.33, respectively. Our adaptive version of FMWM uses different values of α, β, and γ for each assignment with the help of the digital twin running an evolutionary algorithm NSGA-II, detailed in Section 4.7.4

4.7.3 Digital Twin at Design Time

We instantiate *the digital twin layer* of our reference architecture to show how the digital twin is handy at the design stage of LogistiX and particularly to answer RQ4.1 and RQ4.2. As illustrated in Figure 4.3, we deploy a digital twin using Python that simulates task allocations using FMWM, MWM, BG, and FG algorithms. Digital

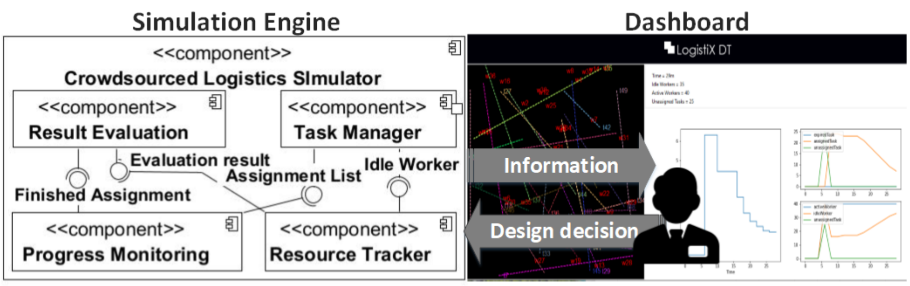
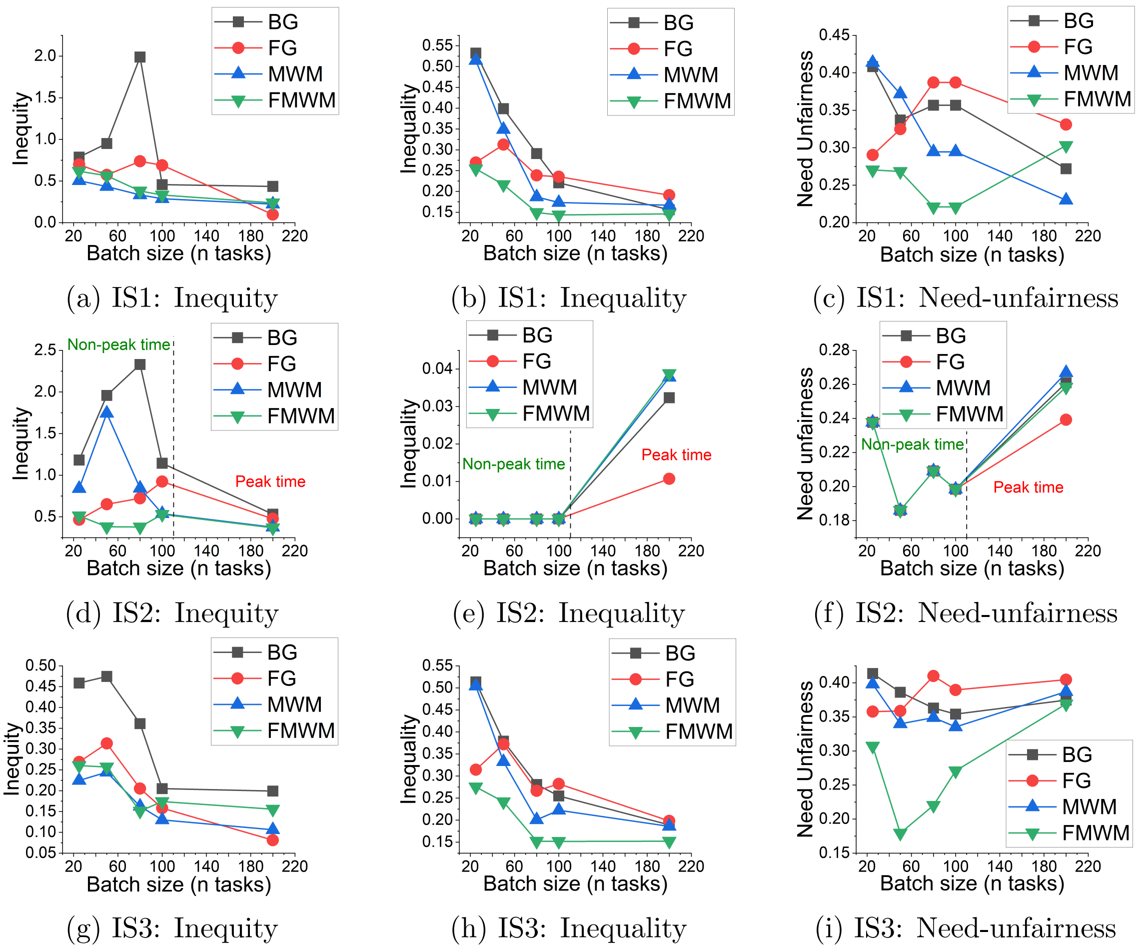


Figure 4.3: Instantiation of the digital twin architecture for assisting the design phase.

twin gives us the flexibility to try these techniques, adopt one suited for a given context and switch among alternatives. The digital twin (i.e., Simulation Engine) sends information to human decision-makers via the dashboard. Our dashboard provides a map of the movement of workers and tasks, along with charts showing statistics.

RQ4.1: To what extent can the digital twin analysis and simulation guide analysts in achieving better fairness for the chosen incentive scenarios?



(a) IS1: Inequity (b) IS1: Inequality (c) IS1: Need-unfairness

(d) IS2: Inequity (e) IS2: Inequality (f) IS2: Need-unfairness

(g) IS3: Inequity (h) IS3: Inequality (i) IS3: Need-unfairness

Figure 4.4: Fairness evaluation employing 100 workers under different incentive scenarios: IS1, IS2, and IS3.

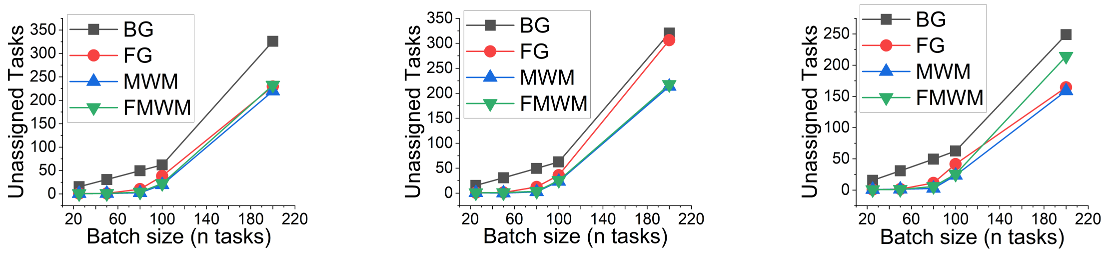
We use the digital twin to test each algorithm in IS1, IS2, and IS3 scenarios using type A datasets with fixed batch sizes for each assignment. Based on the experimental results, we could determine the most suitable algorithm for the runtime.

Scenario IS1 is not designed with fairness in mind. This scenario does not guarantee that the more distance workers travel, the higher their income will be. As shown in Figure 4.4a to 4.4c, as long as the workers-to-tasks ratio is less than or equal to one, FMWM provides significantly lower inequality and need-unfairness than MWM at the little expense of inequity. The greedy-based approaches (i.e., BG, FG) provide such a high level of unfairness, although, in small batch sizes, FG is generally better than BG.

Equality and need are guaranteed in scenario IS2 during non-peak time, as elucidated in Figs. 4.4d to 4.4f. FMWM provides better fairness with the lowest inequality in all batch sizes in this condition. However, FG is much fairer at peak time, even though it allocates fewer tasks than FMWM and MWM.

Since the reward is proportional to the distance, scenario IS3 naturally yields equity. As a result, compared to IS1 and IS2, all algorithms provide smaller inequality in IS3. Figs. 4.4g to 4.4i indicate that FMWM performs better than other algorithms in reducing inequality and need-unfairness but is somewhat worse than MWM in promoting inequity in IS3.

Utilising a digital twin during the design phase lets us comprehend that FMWM is employed more effectively on IS1 and IS3 since it may lessen inequality and need-unfairness rather than only inequity as in IS2. Thus, we utilise scenario IS3 to respond to RQ4.3.



(a) Scenario IS1 (b) Scenario IS2 (c) Scenario IS3

Figure 4.5: Unassigned tasks in different incentive scenarios: IS1, IS2, and IS3.

RQ4.2: How can the analysis assist in a better understanding of the impact of batch size on unfairness indices (in Section 4.4.3) and the number of unassigned tasks?

From our what-if analysis, we understand that the batch size determines the ratio of workers to tasks. The smaller the batch size, the fewer tasks are accessible to workers since we employ a fixed number of workers. Hence, workers' competition grows as batch size lowers, leading to a rise in inequality and inequity in scenarios IS1 and IS3. The need-unfairness tends to diminish as the batch size rises in scenario IS1 but not in scenario IS3. This finding suggests that fairness is affected by the stochastic composition of the workers and tasks on each assignment.

Nevertheless, the pursuit of fairness inevitably comes at a cost – a potential reduction in overall system performance, characterised by its ability to efficiently process all available requests or tasks within the shortest possible time frame. Figure 4.5 illustrates a key observation: some tasks are not allocated and remain in the queue until assigned or expire, significantly when the batch size exceeds the workers' capacity. MWM outperforms FMWM in this context. This issue becomes even more pronounced when the expiry time is shorter, mainly due to the limited number of workers meeting spatiotemporal constraints.

Therefore, it is essential to determine an optimal expiration time for each task, taking into account customer satisfaction. It is also crucial to limit the batch size to the number of available workers and establish the optimal composition and location of the workers' fleet to ensure timely task completion while maintaining fairness.

4.7.4 Digital Twin for Runtime Adaptation

We instantiate our three-layer-based digital twin reference architecture: *digital twin layer, decision layer,* and *asset layer* to demonstrate LogistiX operating in a production environment and to answer RQ4.3, as shown in Figure 4.6. At the *asset layer*, we have a crowdsourced logistics platform simulation that senses task and worker initiation of the dataset. For each assignment, the Task Manager sends a request to *the digital twin layer* to get the value α, β, γ. This request is handled by the Simulation & Analysis component of the digital twin running optimisation using an evolutionary algorithm to generate a Pareto set containing non-dominated solutions of α, β, γ. The digital twin replies to the request with a message containing a knee point defined by an algorithmic decision-maker.

In this instance, we merge the decision layer with the digital twin layer since the decision is determined algorithmically. Given that the evolutionary algorithm is more computationally intensive and requires more resources, the digital twin should be run in a better infrastructure than its assets to ensure stability and scalability. We use ZeroMQ Message Transport Protocol (ZMTP) to provide a communication channel between the digital twin and the asset.

We briefly describe the evolutionary algorithm and knee-point selection as follows.

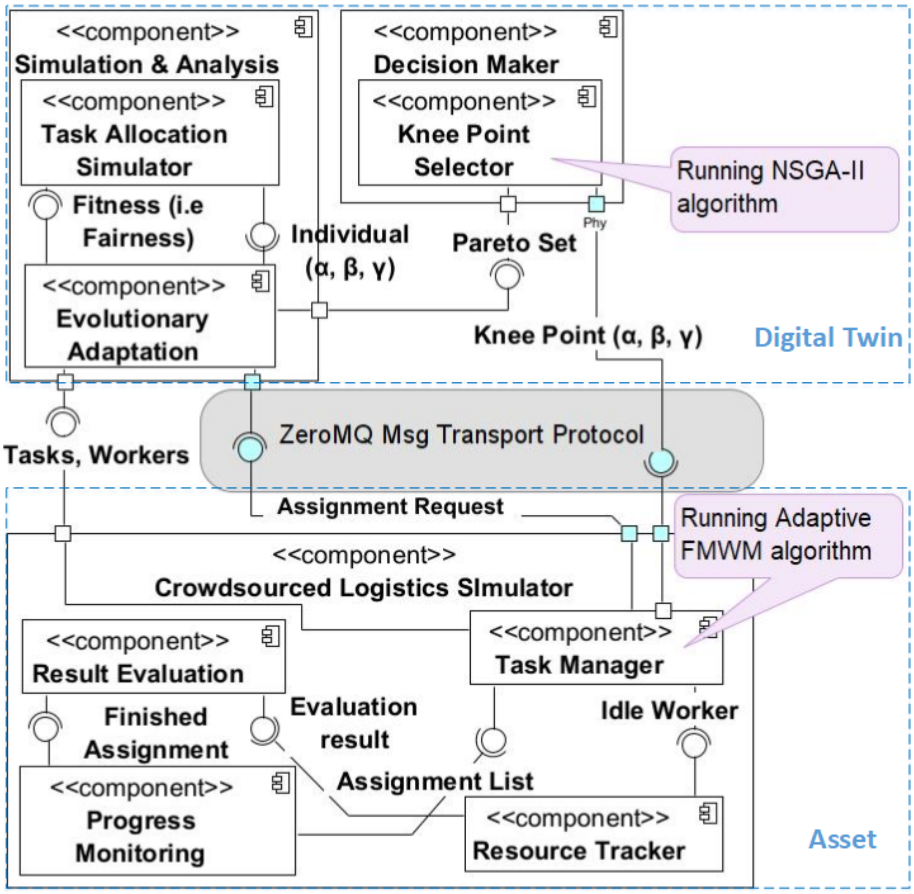


Figure 4.6: Instantiation of the digital twin architecture for runtime adaptation.

Dynamic Evolutionary Approach

Initialization, selection, genetic operators, and termination are the four general phases of an evolutionary algorithm. During the initialization, we use a dual population that combines the pre-defined seed population and a new randomised population to speed up the convergence. We represent each individual's chromosome using three float numbers $[g_1, g_2, g_3]$ where $\sum_{i=1}^{3} g_i = 1, g_i > 0$. The fitness value of each individual is evaluated by running FMWM using its chromosome as α, β, γ, respectively. We employ NSGA-II [181] for the selection and use swap mutation and one-point crossover for the genetic operators. Our approach can find a near-optimal solution with only five generations and a population size of five for the second to fifth generations. The first-generation population comprises ten individuals from the pre-defined seed population plus five new randomised individuals. After the iteration, a knee point is selected from the Pareto fronts described in Section 4.7.4. We find that, in our settings, using larger populations and generations can improve results, but not significantly, with the trade-off of a longer processing time.

Knee Point Selection

All non-dominated solutions in the Pareto set provide trade-offs for each objective. A knee point can be a preferred solution, which suggests an appropriate one. Since we have three objective minimization functions formally given in equation 4.10, we define (0,0,0) as the utopia point. The knee point is the Pareto point closest to the utopia point in a 3D Euclidean space, as illustrated in Figure 4.7b.

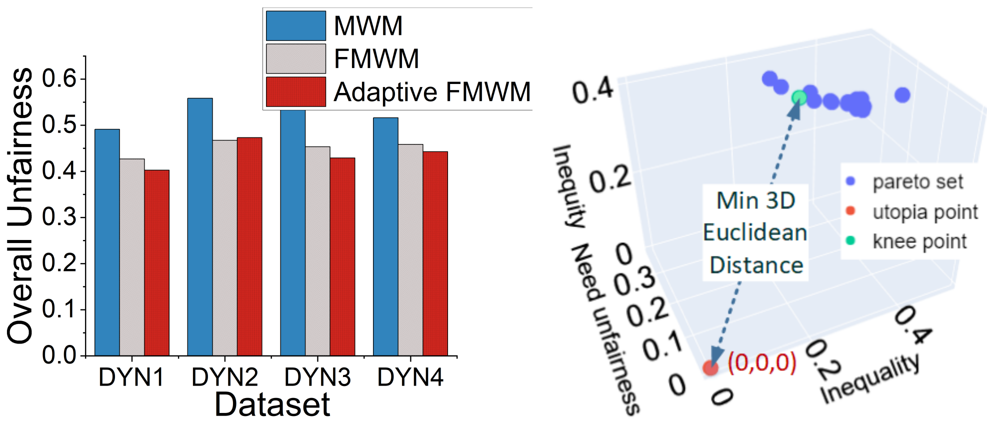


Figure 4.7: (a) Overall unfairness in four datasets with dynamic batch size using scenario IS3 (b) Knee point selection

RQ4.3: How can runtime adaptation, supported by digital twin, better satisfy the fairness requirement and promote social welfare?

We employ four datasets with random batch sizes representing dynamic real-world settings. Figure 4.7a indicates that employing runtime adaptation using digital twin

(Adaptive FMWM) to determine the values of α, β, and γ results in better overall fairness than using static α, β, and γ values (FMWM). The distance to the utopia point is calculated to determine the overall unfairness. This computation is done by analysing all workers' data after the simulation, making it look like FMWM produces slightly better outcomes in DYN2, even though actually Adaptive FMWM produces superior results at the end of each shift.

"In economic terms, social welfare is an aggregation of the welfare or utility of the individual members of society" [182]. Income is a crucial indicator of economic well-being, and it can provide valuable insights into the financial resources available to individuals or groups. Table 4.5 shows that our concepts, FMWM and Adaptive FMWM, have promoted social welfare by ensuring that human car drivers earn more income on average than AVs. Additionally, the average earnings of underprivileged workers (i.e., cyclists and moped drivers) are also higher than in MWM. Social welfare is a multi-dimensional concept that encompasses various aspects of well-being. Relying solely on income and fairness may not provide a complete picture of overall social welfare. However, other dimensions of social welfare are beyond the scope of this research.

Having the digital twin hosted on a better infrastructure provides the crowd-sourced logistics platform with more resources to monitor, track, and receive more requests while waiting for updates from the digital twin. While seed population helps speed up convergence, implementing the digital twin on infrastructure that allows distributed parallel processing seems promising to accelerate evolutionary computing further and is highly recommended for production environments.

Table 4.5: Average earnings per worker on different Type B dataset

Worker type	Average earnings per worker											
	Dataset DYN1			Dataset DYN2			Dataset DYN3			Dataset DYN4		
	MWM	FMWM	Adaptive FMWM	MWM	FMWM	Adaptive FMWM	MWM	FMWM	Adaptive FMWM	MWM	FMWM	Adaptive FMWM
Car driver	2685.24	2873.69	2694.90	2368.52	2654.04	2595.28	2686.51	2603.58	2529.44	2417.47	2524.00	2528.12
Moped driver	2275.93	2177.70	2258.16	1853.61	2039.19	2003.95	1820.57	1993.11	2014.43	2226.94	2398.30	2366.11
Cyclist	917.51	1169.84	1202.78	1067.19	1225.09	1232.87	835.93	1078.66	1127.46	917.85	995.08	991.50
AVs	2733.47	2260.19	2354.79	2331.17	1860.10	1929.77	2418.49	2027.36	2006.16	2351.60	1979.32	2013.08

4.8 Threats to Validity

This chapter may have some potential threats to external validity, particularly in terms of generalisability to real-world applications or different settings. Due to the unavailability of corresponding real-world datasets, our simulation relies solely on synthetic datasets to create a controlled environment.

While we take into account the dynamic aspect of the task-to-worker ratio in the type B datasets, we acknowledge that other dynamic factors, such as task execution failures, varying start and finish shift times, traffic dynamics, etc., are not considered. As a result, the use of synthetic datasets may limit the generalisability of our findings to real-world applications.

Furthermore, the inequity, inequality, and need-unfairness indices that we employ are specifically tailored to our motivating scenario and may not be applicable in a different context, necessitating a redefinition of these metrics. However, our results offer valuable insights within its specific scope and can serve as a foundation for further research or practical applications.

4.9 Conclusion

We have introduced a novel reference digital twin architecture and applied it in real-world scenarios to assess and adapt our bipartite graph matching approach. This endeavour aims to promote equity, equality, and the fulfilment of needs, fostering a more inclusive and sustainable collaboration between humans and machines in crowdsourced logistics. Our approach outperforms baseline methods in terms of overall fairness, as demonstrated through a series of experiments in diverse scenarios. By enhancing fairness between humans and automated systems, our approach establishes a foundation for ethical and effective collaboration within crowdsourced logistics, highlighting its potential for broader applications in sustainable logistics systems and other areas of human-machine collaboration.

Chapter 5

SPECTRA: a Markovian Framework for Managing NFR Tradeoffs with Diverse Levels of Observability

HitLCPS is one of many Self-adaptive systems (SAS) that operate in heterogeneous and dynamic environments, where satisfying multiple quality attributes, also known as NFRs, and resolving dynamic trade-offs among NFRs remains challenging for SAS. However, while many NFRs can be directly observed, some of them are only partially observable (e.g., emotion, job satisfaction, fatigueness, etc.). Existing solutions typically consider homogeneous observability, with all the NFRs under investigation being either fully observable or partially observable. Moreover, existing solutions tend to adopt a single objective approach with scalar rewards, which conceals the individual priorities of each NFR.

This chapter introduces SPECTRA[1]*, a framework based on the Markov decision process (MDP) that utilises vector rewards to explicitly represent multiple NFRs*

[1]The name "SPECTRA" draws inspiration from the quote by Hawking and Mlodinow [183]: "The past, like the future, is indefinite and exists only as a spectrum of possibilities." It reflects the uncertainty that our framework addresses.

and their priorities during decision-making. This framework is designed to handle trade-offs among NFRs in dynamic environments with diverse NFRs observability. Additionally, this chapter also presents MR-MOMDP, which goes beyond MDP approaches to incorporate multiple objectives in a mixed-observability setting. The evaluation uses hypothetical remote data mirroring (RDM) scenarios on a non-trivial scale using digital twin architecture to demonstrate the effectiveness of our approach in dynamically managing tradeoffs in SAS in the presence of mixed NFRs observability. The results indicate that our approach can achieve higher utility values, reduce the time needed for policy planning, and better satisfy the NFRs. This chapter is adapted from [31].

5.1 Introduction

Self-adaptive systems (SAS) must adapt to satisfy functional and non-functional requirements in dynamically changing environments that often involve tradeoffs [184]. For example, while encrypting data streams can guarantee data confidentiality, it may also decrease performance, or while implementing the heartbeat protocol that enables monitoring of availability, excessive usage can negatively impact performance and bandwidth efficiency [185]. Therefore, effectively managing tradeoffs in SAS presents a complex challenge, as it poses the need to find a delicate balance while accommodating the environment's dynamic nature through context-specific configurations [184]. Additionally, SAS must navigate uncertainties arising from diverse sources, including external factors and internal intricacies within the software systems [114]. Decisions regarding adaptation should be guided by the priorities of the NFRs [29]. Without understanding the priorities of each NFR, it is difficult to adequately deal with the tradeoffs between conflicting objectives. Therefore, it is crucial to understand stakeholders' preferences and take into account the NFR's priorities to facilitate decision-making at runtime.

To ensure effective NFRs satisfaction evaluation, it is advisable to define the measurable metrics of the NFRs [186]. For instance, performance can be evaluated through time measurements; efficiency is commonly linked to measurable resource consumption, and so forth. However, there may be instances where direct observability is not possible, and therefore partial observability [2] is unavoidable. This is often the case in human-machine collaboration, where while states related to machine characteristics are usually fully observable, states related to human traits are commonly only partially observable. Other examples are demonstrated in the motivating scenario in Section 5.4, which concerns a Remote Data Mirroring (RDM) system [187, 188].

Existing SAS solutions have modelled their domain problems as sequential stochastic decision-making and control. Earlier approaches [124, 125, 29] have relied on Partially Observable Markov Decision Processes (POMDPs) when it is not feasible to directly observe all metrics associated with NFRs. Conversely, when the satisfaction of NFRs can be directly observed, Markov Decision Processes (MDPs) have been employed [113]. Some of these approaches use a single-objective technique that employs scalar rewards to represent a combined cardinal priority for all NFRs [113, 124, 125].

Nevertheless, we argue that leveraging existing POMDP and MDP approaches to address challenges in SAS, which can be characterised by mixed observable NFRs, may not yield optimal outcomes or efficient processing time. Neglecting partial observability within an MDP framework can lead to imbalanced decisions, favouring fully observable NFRs to the detriment of partially observable ones or vice versa. On

[2]The term “observability” comes from control theory, defined as the degree to which one can comprehend the internal state or condition of a complex system based only on knowledge of its external outputs [123].

the other hand, modelling all aspects as partially observable through POMDP incurs a larger belief space, often culminating in fuzzy tradeoff resolutions that might not fit the problem under investigation. The POMDP approach could also lead to extended processing times compared to using Mixed Observable Markov Decision Process (MOMDP) models [128]. Employing scalar rewards in single-objective approaches conceals the impact of adaptation on the individual satisfaction and priorities of each NFR. Using vector reward functions in SAS offers a more explicit representation of the priorities assigned to different NFRs during runtime, thereby enabling informed decision-making [29].

In a nutshell, this chapter makes the following novel contributions:

Firstly, it goes beyond existing work by proposing a solution that factors in heterogeneous setups of NFRs observability, whether fully or partially, when satisfying them and dynamically managing their conflicts under uncertainty. Our proposed framework, called SPECTRA, serves as a guideline to assist designers and software engineers of SAS in deciding on the suitable Markov model, informed by the observability of the NFR metrics and uncertainty in satisfying the NFRs and managing their tradeoffs.

Secondly, our approach surpasses the traditional MOMDP model by considering multiple objectives in a mixed observability environment. We have developed the Multi-Reward MOMDP (MR-MOMDP), which utilises vector rewards instead of scalar rewards. The MR-MOMDP model uses OLSAR-Perseus [189] to identify a subset of Pareto optimal solutions. To our knowledge, our research is the first to explore multi-objective (i.e., multi-reward) MOMDP. The MR-MOMDP model can be applied to various SAS application domains, including CPS, IoT, robotics, etc., where multi-objective problems and mixed observability are present.

Thirdly, the proposed approach is implemented as a publicly accessible Python framework, extending the POMDPy framework [190]. This extension incorporates additional features, including model parsers to cater to different model specification format, portable policy generation, which support different applications, and solvers to support both single and multi-objective POMDP and single and multi-objective MOMDP, expanding the capabilities of the original framework, initially designed for single-objective POMDP only.

Fourthly, the approach is evaluated using a non-trivial case of a Remote Data Mirroring system [187, 188]. The results show the potential of our method for controlling tradeoffs dynamically in SAS while dealing with mixed NFR observability. To assist software engineers and SAS designers in experimenting with and potentially using our approach, we have provided the model, solvers, and policy generators as an artefact for adoption and further application.

5.2 Background

In this section, we provide a brief overview of MDP, POMDP, MOMDP, and MR-POMDP, which are the foundational concepts of this research.

5.2.1 MDP, POMDP, and MOMDP

The Markov Decision Process (MDP) [191] is a mathematical framework employed in decision-making, structured as a series of states, actions, and rewards. At each timestep, the agent determines its current state and selects an action, which results in a reward, based on the action taken and the resulting state change. The objective of an MDP is to identify the optimal policy, which specifies the best action for each state to maximise the total expected rewards over time.

The Partially Observable Markov Decision Process (POMDP) [192] is an extension of the Markov Decision Process (MDP) where the state of the system cannot be directly observed or is partially observable. Partial observability can be either *intrinsic* attributed to the innate nature of the NFR and the problem where observability is difficult (e.g., reliability [193], emotion, satisfaction, etc.), or *extrinsic*, where the environment (e.g., sensor inaccuracy and other technical constraints) makes it difficult to directly measure NFR metrics, despite them being actually fully observable by nature (e.g., network security [193]). Unlike MDP, which assumes full observability of states, POMDP deals with partial observability using belief states. This makes POMDP more challenging because it has to find an optimal policy with an uncertain actual state of the system, relying on observations.

The Mixed Observable Markov Decision Process (MOMDP) [128] framework generalises POMDP, allowing for a combination of fully observable and partially observable states in the model. The decision-making agent considers both fully and partially observable aspects of the environment when choosing an appropriate action based on the underlying state.

The multi-reward (MR) version of MDP, POMDP, and MOMDP employs vector rewards instead of scalar rewards. The vector reward dimensions represent the number of objectives (i.e., NFRs to satisfy). Figure 5.1 illustrates MR-MDP, MR-POMDP, and MR-POMDP with vector rewards for three objectives problem. We discuss more on MR-MOMDP in section 5.3, and MR-POMDP in section 5.2.2 below.

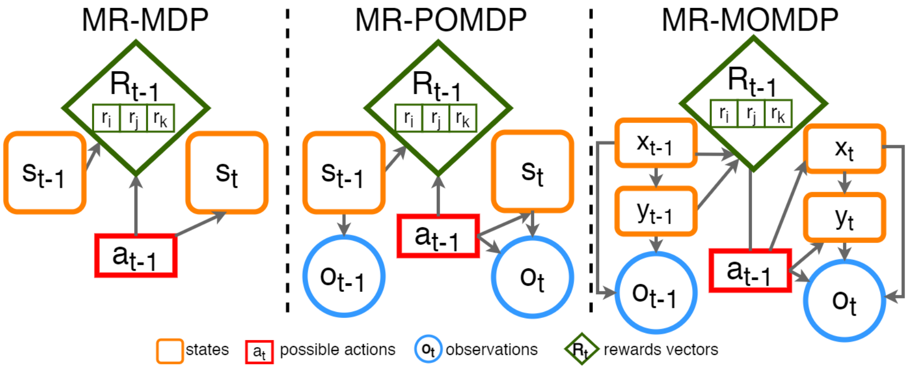


Figure 5.1: MR-MDP, MR-POMDP, and MR-MOMDP model

5.2.2 Multi-Reward Partially Observable Markov Decision Process (MR-POMDP)

An MR-POMDP [30, 189] model is a tuple $\{S, A, \Omega, b, T, O, R, \gamma\}$ where S represents the set of discrete states of the environment which are not directly observable to the agent; A denotes a finite set of all possible actions that the agent can take; Ω represents the set of discrete observations that the agent can receive; The initial belief state, denoted by b, is the probability distribution over S; $T(s, a, s')$ is the transition function that defines state-to-state transition probabilities; $O(s', a, o)$ is observation function that generates observation outcome probabilities; $R(s, a)$ is the reward function defining the reward vector given after doing an action a in the state s, given by $R(s, a) = [r_1, \ldots, r_k]$, where k is the number of objectives; and γ is the discount factor, which ranges from 0 to 1, used to prioritise immediate rewards over future rewards.

At each timestep, the agent receives a noisy/partial observation $o \in \Omega$ from the environment. The outcome of observing o is given by $O(s', a, o) = p(o|s', a)$ for $s' \in S$ and $a \in A$. The agent selects an action a based on the observation result and its belief state b to move from the current state s to the next state s'. The next state s' is given by the transition function $T(s, a, s')$ that specifies the probability distribution over the next state s' given the current state s and action a. The agent receives an immediate reward defined by reward function $R(s, a)$ for doing action a in state s. The environment generates new observations the agent uses to update its belief state using Equation 5.1:

$$b'(s') = \frac{O(s', a, o) \sum_{s \in S} T(s, a, s') b(s)}{p(o|b, a)}, \tag{5.1}$$

where $p(o|b, a) = \sum_{s' \in S} O(s', a, o) \sum_{s \in S} T(s, a, s') b(s)$ is the normalisation constant.

An MR-POMDP plan is called a policy, denoted by $\pi(b) : B \mapsto A$ takes the belief state $b \in B$ and weight vector w as input and returns the action $a \in A$ to be executed by the agent. The agent's goal is to find an optimal policy π^* whose scalarised value satisfies the Bellman optimality equation, defined in Equation 5.2:

$$V^*(b, w) = \max_{a \in A} \left[\sum_{s \in S} R(s, a).w.b(s) + \gamma \sum_{o \in \Omega} p(o|b, a) V^*(b', w) \right] \tag{5.2}$$

V^* can be approximated by iterating a number of stages where a value function V_n at stage n is parameterized by a finite set of α-matrices $\Theta_n = \{\alpha_n^i\}, i = 1, \ldots, |V_n|$. Each α-matrix α_n^i is associated with an action a. It comes with $|S| \times k$ dimension, where k is the number of objectives, as in Equation 5.3. Each row represents a vector value for each state s.

$$\alpha_n^i = \begin{bmatrix} Obj_1 & \dots & Obj_k \\ V(s_1)_1 & \dots & V(s_1)_k` \\ \vdots & \ddots & \vdots \\ V(s_j)_1 & \dots & V(s_j)_k \end{bmatrix}, \tag{5.3}$$

where $V(s_j)_k, s \in S, k \geq 1, j = 1, \dots, |S|$, is the value of state s_j at k-th objective.

The value function of a belief b at stage n is the inner product of belief b with the α-matrix $\alpha_n^i \in \Theta_n$ and weight vector w given by Equation 5.4:

$$V_n(b, w) = \max_{\alpha_n^i \in \Theta_n} b.\alpha_n^i.w \tag{5.4}$$

A point-based solver uses approximate backups to calculate the optimal α-matrix for a selected set B of sampled belief points. A point-based backup [189] is started by determining the back-projection $g_i^{a,o}$ of each next-stage value matrix $\alpha_i \in \Theta_n$ for every action a and observation o, given by Equation 5.5.

$$g_i^{a,o} = \sum_{s' \in S} O(a, s', o) T(s, a, s') \alpha_i(s') \tag{5.5}$$

Then, for each $b \in B$, the back-projection matrix $g_i^{a,o}$ is used to construct a new set of α-matrices, given by Equation 5.6 where r^a is a rewards matrix for performing action a in every state. Each α-matrix α_{n+1}^a corresponds to an action $a \in A$, so in total, there will be $|A|$ number of new α-matrices.

$$\alpha_{n+1}^a = r^a + \gamma \sum_{o \in \Omega} \arg\max_{g^{a,o}} b.g^{a,o}.w \tag{5.6}$$

Finally, the new optimal α-matrix for b, denoted as $\alpha_{n+1}^{b,a}$, is the α_{n+1}^{a} that maximises $V_{n+1}(b,w)$ (cf. Eq. 5.4):

$$\alpha_{n+1}^{b,a} = \arg\max_{\alpha_{n+1}^{a}} b.\alpha_{n+1}^{a}.w \tag{5.7}$$

Given a belief b and weight vector w, a policy $\pi(b,w)$ at a particular stage $n+1$ is defined by choosing an action a that corresponds to the α-matrix $\alpha_{n+1}^{b,a}$.

Expanding upon the foundational concepts, in section 5.3, we present our proposed approach of MR-MOMDP that deals with the mixed observability of the NFRs along with modelling of the individual priorities of NFRs to support the decision-making of SAS.

5.3 Multi-Reward Mixed Observable Markov Decision Process (MR-MOMDP)

In this section, we introduce the MR-MOMDP approach, which exhibits several shared attributes with MR-POMDP.

5.3.1 Model

A MOMDP [128] is a type of POMDP where the environment includes states directly observed by the agent and partially observable states. Similar to MR-POMDP, our proposed MR-MOMDP utilises a reward vector in which each axis component of the reward vector reflects a goal (e.g., NFR satisfaction level). The MR-MOMDP is defined as a tuple $\{X, Y A, \Omega, b, T_x, T_y, O, R, \gamma\ \}$. The set X contains fully observable states $x \in X$, while the set Y contains partially observable states $y \in Y$. A complete system state is represented by a tuple (x, y), so the state space is factored as $S = X \times Y$. To transition from a start state (x, y) to an end state (x', y'), the agent takes an action a, and the value of x' is determined by the transition function $T_x(x, y, a, x') = p(x'|x, y, a)$, while the value of y' is determined by the transition function $T_y(x, y, a, x', y') = p(y'|x, y, a, x')$. The remaining components of the MR-MOMDP are the same as those in the MR-POMDP, including the set of observations Ω, the observation function O, the vector reward function R, and γ, which is the discount factor with a real value $\in [0, 1]$.

By decomposing the state space into fully and partially observable states, expressing the belief space B in a factorised manner becomes feasible, reducing its high dimensionality [128]. As a result, any belief b in MR-MOMDP regarding the complete system state $s = (x, y)$ can be represented as (x, b_y), where $b_y \in B_y$ denotes the belief concerning the partially observable state y and B_y denotes the space of all beliefs on the partially observable state variable y. Each B_y is associated with the fully observable state x, denoted as $B_y(x)$. Therefore, as B is a union of $B_y(x)$ for $x \in X$, B possesses $|X| \times |Y|$ dimensions, and each $B_y(x)$ has $|Y|$ dimensions only, where $|X|$ and $|Y|$ indicate the number of fully observable and partially observable states in X and Y, respectively.

Unlike in an MR-POMDP, each fully observable state x in MR-MOMDP maintains its α-matrices set, denoted as $\Theta_y(x)$, defined over $B_y(x)$. This approach divides the belief space into smaller subspaces that are easier to manage and analyse, making it more efficient than representing the whole belief space as one complex entity [128].

$$V((x, b_y), w) = \max_{\alpha \in \Theta_y(x)} b_y.\alpha.w \tag{5.8}$$

To compute the value $V((x, b_y), w)$ in equation 5.8, we initially utilize the value of x as an index to locate the corresponding α-matrices set and subsequently determine the maximum α-matrix within that set using equation 5.9.

$$\alpha = \arg \max_{\alpha \in \Theta_y(x)} b_y.\alpha.w \tag{5.9}$$

5.3.2 Solver

We employ Optimistic Linear Support with Alpha Reuse (OLSAR) [189], a point-based solver used in MR-POMDP [189, 29], and modify it to fit the MR-MOMDP. OLSAR is a method to solve multi-objective decision problems using a series of calls of a bounded approximate single-objective POMDP solver (e.g., *OCPerseus* [189]) that leverages previously discovered α-matrices to create an initial lower bound for subsequent OCPerseus calls, enabling faster resolution of the scalarized POMDP.

Our MR-MOMDP solver consists of three layers. The first layer is OLSAR, an outer loop that calls *OCPerseus* on the second layer for each iteration until the convergence criteria are reached. The third layer is a multi-objective backup function (i.e., *backupMO*) that implements back-projection.

Algorithm 2: OLSAR(b_0, η, ϵ) for MR-MOMDP, adapted from [189].

Input : Initial belief $b_0 = (x_0, b_{y0})$, Convergence threshold η, Max. allowed error ϵ

Output : A set of partial approximate CCS of multi-objective value vectors C_v and its associate weights set C_w

$C_v \leftarrow \emptyset$ `// A set of partial approximate CCS of multi-objective value vectors` V_{b_0}
$C_w \leftarrow \emptyset$ `// A set of weights associated with every value in` C_v
$Q \leftarrow \emptyset$ `// A queue of weights to search with their priority` $\{(w, \Delta_w)\}$
Add extrema weights to Q with infinite priority
Initialise Θ_{all} `// A set of` α`-matrices that create a minimum estimate of the value.`
Initialise B `// A set of sampled belief points obtained by random exploration`
while $\neg Q.isEmpty() \wedge \neg timeOut()$ **do**
  $w \leftarrow Q.dequeue()$ `// retrieve weight vector`
  $\Theta_r \leftarrow$ select the best α-matrix α from Θ_{all} for each $(x, b_y) \in B$, given w
  $\Theta_w \leftarrow$ **OCPerseus**(Θ_r, B, w, η)
  $V_{b_0} \leftarrow \max_{\alpha \in \Theta_w} b_{y0}.\alpha.w$
  $\Theta_{all} \leftarrow \Theta_{all} \cup \Theta_w$
  if $V_{b_0} \notin C_v$ **then**
    $W_{del} \leftarrow$ delete the corner weights that V_{b_0} made obsolete from Q.
    $C_v \leftarrow$ remove all values made obsolete by V_{b_0} from C_v
    $C_v \leftarrow C_v \cup V_{b_0}$
    $C_w \leftarrow C_w \cup w$
    $W_{del} \leftarrow W_{del} \cup w$
    $W_{corner} \leftarrow$ newCornerWeights(V_{b_0}, W_{del}, C_v)
    foreach $w \in W_{corner}$ **do**
      $\Delta_w \leftarrow$ calculate max. possible improvement (i.e., priority) using (w, C_v)
      if $\Delta_w > \epsilon$ **then**
        Q.add($\{(w, \Delta_w)\}$)
      end
    end
  end
end
return C_v, C_w

Layer 1. OLSAR for MR-MOMDP

OLSAR, as presented in Algorithm 2, takes initial belief b_0 as a tuple (x_0, b_{y0}), a convergence threshold η, and maximum allowed error ϵ to produce a set of partial approximate convex coverage sets (CCS) of multi-objective value vectors, denoted as C_v, and its associate weights set C_w. It begins by initialising a priority queue Q and assigning extrema weights to Q with infinite priority. It also initialises a set of α-metrices Θ_{all} that define a minimum of the value and a set of sampled belief points B that can be obtained through random exploration. Θ_{all} consists of $|X|$ subsets where each subset corresponds to a fully observable state variable x.

While Q is not empty and a timeout threshold has not been reached, OLSAR does the following: it retrieves the weight vector w from Q and given w, it selects the best α-matrices from Θ_{all} for each $(x, b_y) \in B$ and stores them into $\Theta_r = \{\Theta_{r_y(x)} | \Theta_{r_y(x)} = \{\alpha\}, x \in X\}$. It then calls $OCPerseus(\Theta_r, B, w, \eta)$ that solves MR-POMDP through scalarisation using w to obtain new α-matrices set Θ_w. OLSAR then updates Θ_{all} with the new Θ_w. Given the initial belief $b_0 = (x_0, b_{y0})$ and Θ_w, a scalarised value of b_0, denoted as V_{b_0}, can be calculated. If V_{b_0} is not in the CCS C_v, it removes all values made obsolete by V_{b_0} from C_v, updates C_v with V_{b_0}, and stores its associated weight vector w into C_w. Then, OLSAR deletes the obsolete corner weights from Q and stores them into W_{del}. Given V_{b_0}, W_{del}, and C_v, OLSAR calculates new corner weights and maximum possible improvements (i.e., priority) Δ_w. For each weight w in new corner weights, If Δ_w is bigger than the maximum allowed error ϵ, OLSAR adds w and its priority Δ_w in the queue Q. These steps are repeated until Q is empty.

Layer 2. OCPerseus

Perseus is a point-based solver, originally for POMDP, that approximates the exact solution using a set of randomised points representing the belief state. This approach allows Perseus to focus on several points of the belief space rather than dealing with a continuous representation of the entire space.

OCPerseus, described in Algorithm 3, takes a set of α-matrices Θ, a set of sampled belief points B, weight vector w, and a convergence threshold η as input to produce a new set of α-matrices Θ'. As explained in the last paragraph of section 5.3.1, both Θ and Θ' consist of $|X|$ subsets of α-matrices where each subset corresponds to a fully observable state variable x.

OCPerseus performs several backup stages until it reaches the convergence threshold η. The backup stage begins by setting Θ' to an empty set and initialising B' by copying B to B'. B' is used to keep track of non-improved belief points consisting $b = (x, b_y) \in B$ whose $V_{n+1}(b, w)$ is still lower than $V_n(b, w)$ (line 11). As long as B' is not empty, *OCPerseus* randomly selects a belief point (x, b_y) from B' (line 8). Given b, w, and Θ, if possible, an improving α-matrix α is selected and added to Θ. Then Θ' is updated with the best α-matrix from Θ. Next, all the improved belief points are deleted from B'. These steps are repeated until B' is empty.

Layer 3. backupMO for MR-MOMDP

Since, in MR-MOMDP, each belief point $b \in B$ is a tuple (x, b_y), each observable state variable $x \in X$ maintains the back-projection matrix $g_i^{x,a,o}$ of next-stage value

Algorithm 3: OCPerseus(Θ, B, w, η) adapted from [189].

Input : A set of α-matrices Θ, A set of belief points B, weight vector w, Convergence threshold η

Output : A new set of α-matrices Θ'

```
Θ' ← Θ
Θ ← {−∞⃗}
while max_{b∈B} ( max_{α'∈Θ'} b.α'.w − max_{α∈Θ} b.α.w ) > η do
    Θ ← Θ'
    Θ' ← ∅
    B' ← B
    while B' ≠ ∅ do
        b ← randomly select a belief point b = (x, b_y) from B'
        α ← backupMO(Θ, b, w)  // select α-matrix
        Θ' ← Θ' ∪ {arg max_{α'∈(Θ∪α)} b.α'.w}
        B' ← {b | b ∈ B', max_{α'∈Θ'} b.α'.w < max_{α∈Θ} b.α.w}
    end
end
return Θ'
```

matrix $\alpha_i \in \Theta_y(x)$, formulated in equation 5.10.

$$g_i^{x,a,o} = \sum_{x' \in X} \sum_{y' \in Y} O(a, x', y', o) T_x(x, y, a, x') T_y(x, y, a, y') \alpha_i(y') \tag{5.10}$$

For every $b_y \in B_y(x)$, the back-projection matrix $g_i^{x,a,o}$ is utilized to create a fresh collection of α-matrices, given in Equation 5.11, for each $x \in X$ and for each action $a \in A$. Consequently, the total number of sets of α-matrices will equal $|A| \times |X|$.

$$\alpha_{n+1}^{x,a} = r^{x,a} + \gamma \sum_{o \in \Omega} \arg\max_{g^{x,a,o}} b.g^{x,a,o}.w \tag{5.11}$$

The new optimal *alpha*-matrix for (x, b_y) returned by *backupMO* is given by Equation 5.12.

$$backupMO(\Theta_y, (x, b_y), w) = \arg\max_{\alpha_{n+1}^{x,a}} b_y.\alpha_{n+1}^{x,a}.w \tag{5.12}$$

5.4 Motivating Scenario

Let us consider **MirrorNet**, a hypothetical Remote Data Mirroring (RDM) system that uses Software-Defined Networking (SDN) [194] to allow easy network topology changes without requiring hardware adjustments. RDM is a reliable method for handling failures by making an identical copy of data from a primary storage system or server at a remote location, usually over a network connection. The main purpose of remote data mirroring is to ensure data redundancy, disaster recovery capabilities, and business continuity.

RDM system comprises vital components, such as 1) the primary system, which is the primary data source that requires mirroring. This can be a storage device, a server, or even an entire data centre; 2) remote systems or mirrors are locations where mirrored data is stored and are typically geographically separated from the primary system to minimise risks from regional disasters; 3) mirroring processes/protocols that replicate data from the primary system to the remote system continuously or periodically, synchronously or asynchronously; 4) a network infrastructure provides connections for primary and secondary systems, which can involve different types of wired and wireless networks with varying characteristics.

To establish a network of mirrors and distribute data efficiently across them, MirrorNet employs two different topologies that can be switched dynamically: Minimum Spanning Tree (MST) and Redundant Topology (RT). It has been observed on the MirrorNet that the MST topology requires fewer active network links, reducing bandwidth consumption and operational costs. However, it may be less reliable. Meanwhile, the RT topology requires more active network links, which enhances reliability but would incur higher operating costs due to increased bandwidth consumption. MirrorNet aims to achieve the optimal balance between improved performance at a lower cost (i.e., efficiency) and potential data loss (i.e., reliability) in different situations, including normal and abnormal conditions such as during attacks or anomalies.

Performance can be directly measured based on the writing time, which represents the duration required to copy data to all mirrors, and in this aspect, both MST and RT are comparable. *Efficiency* can be determined by the total bandwidth consumption, which significantly impacts costs. On the other hand, active links and connected mirror nodes do not guarantee network *reliability* because security threats and failures can occur at all layers of RDM, involving various hardware and software components. Additionally, Intrusion Detection System (IDS) sensors have limitations in detecting and mitigating all types of attacks and failures. Nevertheless, we can observe reliability through probability-based metrics [193] such that higher reliability can be observed when there is a greater number of active network links and a lower severity level of attacks.

The following sections discuss how SPECTRA can help MirrorNet manage NFR tradeoffs at runtime with mixed observable metrics.

5.5 The SPECTRA Framework

Our proposed framework, SPECTRA, as shown in Figure 5.2, aims to assist SAS in satisfying their NFRs in environments with many uncertainties. The word "SPECTRA" abbreviates specification and vector reward, which are the heart of our methodology. SPECTRA is based on Markov Decision Process models, which consider *action, state,* and *observation* uncertainty. *Action uncertainty* refers to uncertain consequences of actions; *state uncertainty* pertains to a lack of knowledge about the

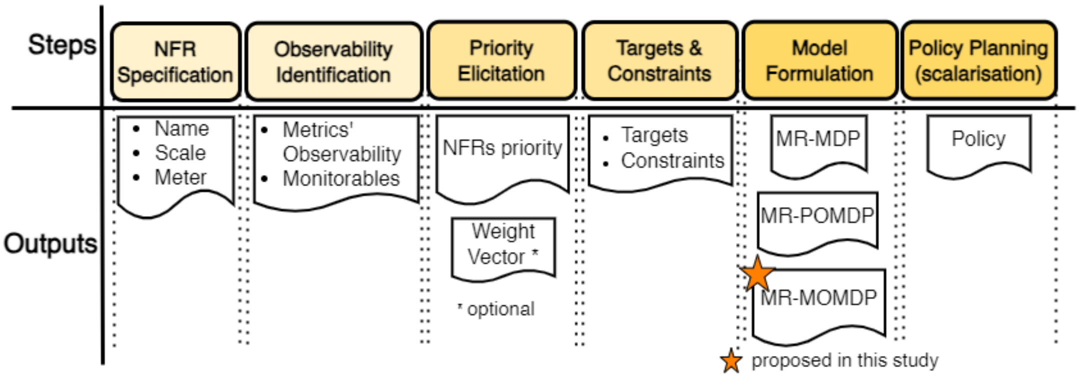


Figure 5.2: The proposed SPECTRA Framework

actual system state; and incomplete, noisy, or ambiguous observation results cause *observation uncertainty*.

This framework serves as a versatile set of steps that can be customised to suit particular domains by employing appropriate techniques. It provides a foundational structure that can be adjusted and fine-tuned according to the unique requirements and characteristics of different contexts and domains. Although the framework is presented as sequential steps/phases, it allows for flexibility in refining the design and model by revisiting previous steps or jumping further ahead. In the following sections, we describe the different phases of SPECTRA by using the hypothetical scenario of MirrorNet, described in section 5.4, as an example.

5.5.1 Phase 1. NFR Specification

At this stage, adopted from the SAFe framework [186], the stakeholders and designer specify NFRs by considering three attributes. Firstly, a *name* is given to each NFR, which may pertain to the system's attributes such as configurability, performance, security, and reproducibility, among others [195]. Secondly, they determine the *scale*, which identifies the object being measured and the corresponding unit of measurement. Finally, they establish the *meter*, which is the measurement method.

Table 5.1: Phase 1 - NFRs Definition

Name	**Scale**	**Meter**
Minimise Cost Efficiency **(MC)**	Bandwidth consumption	Average bandwidth consumption
Maximise Performance **(MP)**	Writing time	Average writing time
Maximise Reliability **(MR)**	Network reliability	Average number of healthy links

These details are outlined in Table 5.1, providing an overview of the NFRs of MirrorNet and their corresponding attributes in phase 1. This approach ensures that the NFRs are well-defined and accurately measured.

Stakeholders and designers must come to a consensus on specifying the NFRs. Who is involved depends on the organisation's policies and the impact of the NFRs. If necessary, they should reconvene to refine and revise the NFRs in response to market dynamics, changes in organizational goals, advancements in technology, and other factors. Consequently, if these changes to the NFRs cannot be accommodated within the current model, a new model is required to retrain the system and generate new policy.

5.5.2 Phase 2. Observability Identification

During this stage, stakeholders and designers assess the *observability* of each metric associated with the NFRs. Observability is categorised as either direct or indirect. If the metric can be directly observed, the NFR is fully observable. On the other hand, if the metric can only be probed through various monitorable objects, it implies that the NFR is partially observable. In the case of direct or fully observable metrics, the metric itself serves as the *monitorable*. Conversely, different objects are used as monitorables to probe the desired metric for indirect or partially observable metrics.

In Table 5.2, MC and MP metrics are easily observable. Bandwidth consumption (BWC) and writing time (WRT) can be directly monitored to determine their

Table 5.2: Phase 2 - Observability Identification

NFR	**Metrics**	**Observability**	**Monitorables**
MC	Average bandwidth consumption	Direct	Bandwidth consumption **(BWC)**
MP	Average writing time	Direct	Writing time **(WRT)**
MR	Average number of healthy links	Indirect	Active network links **(ANL)**, Network attack and anomaly **(ATK)**

averages. However, the number of healthy links cannot be directly observed, as compromised hosts may be connected to active links and exhibit malicious behaviour at the upper layer. Nonetheless, MirrorNet employs a traffic monitoring system that can detect anomalies and network attacks, enabling it to probe the number of healthy links by monitoring the active network links (ANL) and the status of network attacks and anomalies (ATK).

5.5.3 Phase 3. Priority Elicitation

Preference elicitation constitutes a foundational issue in creating intelligent systems that make or recommend decisions on behalf of users[196]. Priorities and preferences need to be elicited from domain experts and stakeholders, which serve as guiding factors for decision-making in adaptation, aligning with the system's objectives [197]. Various decision-making techniques (e.g., AHP [198], TOPSIS [199], etc.) and utility theories [200, 201] can be employed in this phase.

Stakeholders may specify the tradeoff between NFRs by creating a utility function that assigns weight to each quality attribute, i.e., forming a weight vector (e.g., [0.333, 0.333, 0.333]), incorporating quality attribute scenarios, or utilizing conditional logic [185].

If a global weight vector is provided, we do not need to find the set of optimal weights in Phase 6 - Policy Planning. Policies can be generated by scalarising and solving the model using a single-objective solver. In the case of MirrorNet, priorities are not elicited as a weight vector but rather as a conditional logic, as stated below:

> ”In normal to moderately detrimental conditions, priority is given to cost efficiency and performance, with reliability being less important. However, in highly severe detrimental conditions, reliability holds the highest priority, resulting in a relaxation of cost efficiency and performance requirements.”

The priorities and preferences generated in this phase determine how targets and constraints can be determined in the subsequent phase, as well as the reward function of the model.

5.5.4 Phase 4. Targets and Constraints

The next stage involves defining *target* values to achieve and *constraints* to avoid, considering the priority given in the previous stage. This step is crucial in establishing clear objectives and boundaries that determine the satisfactory level of the system.

> “If you can’t measure it, you can’t manage it.” - Peter Drucker

To define targets and constraints, our reference values are monitorables rather than metres or metrics, as meters/metrics may not always be directly observable, while we can determine metric values through monitorables. Additionally, we can control the values of monitorables through each action or decision we take in each state, which will impact the monitorables value.

Table 5.3: Phase 4 - Targets in environment SL1 and SL2.

NFR	Targets
MC	Average bandwidth consumption SHALL below the threshold and AS FEW AS POSSIBLE
MP	Average writing time SHALL below the threshold and AS FEW AS POSSIBLE
MR	Average number of active links SHALL above the threshold and AS MANY AS POSSIBLE

Table 5.4: Phase 4 - Constraints for normal and detrimental conditions of environment SL1.

Condition	Normal / Non-detrimental	Detrimental / Severity Level: Low-Moderate
NFR	**Constraints**	**Constraints**
MC	Total bandwidth threshold violations $\leq$ 15%	Total bandwidth threshold violations $\leq$ 15%
MP	Total writing time threshold violations $\leq$ 10%	Total writing time threshold violations $\leq$ 10%
MR	The number of active network links threshold violations $\leq$ 30%	The number of active network links threshold violations $\leq$ 35%

By defining targets and constraints, stakeholders can effectively communicate their expectations and align the system with organisational goals. However, it is possible to relax the target and constraints under particular circumstances to enable SAS to keep functioning in changing and uncertain environments while still meeting critical requirements [202].

Table 5.3 lists the targets of MirrorNet for two different environmental scenarios: SL1 and SL2. The targets are specified using the RELAX language [202] to explicitly acknowledge and address the inherent uncertainty in SAS. The SL1 environment anticipates detrimental conditions with low to moderate severity levels, while the SL2 environment anticipates detrimental conditions with high severity levels.

Table 5.4 and Table 5.5 illustrate the constraints for the SL1 and SL2 environments in both normal and detrimental conditions. Under normal (non-detrimental)

Table 5.5: Phase 4. Constraints for normal and detrimental conditions of environment SL2.

Condition	**Normal / Non-detrimental**	**Detrimental / Severity Level: High**
NFR	**Constraints**	**Constraints**
MC	Total bandwidth threshold violations $\leq 15\%$	Total bandwidth threshold violations $\leq 30\%$
MP	Total writing time threshold violations $\leq 10\%$	Total writing time threshold violations $\leq 10\%$
MR	The number of active network links threshold violations $\leq 30\%$	The number of active network links threshold violations $\leq 5\%$

conditions, both environments share the same constraints, allowing for a higher threshold violation of active network links (ANL) to achieve better performance (MP) and cost efficiency (MC). However, in detrimental conditions, the SL1 environment relaxes its ANL threshold without sacrificing performance (MP) and efficiency (MC) because stakeholders remain confident in the reliability of the MirrorNet infrastructure under low to moderate severity levels. In contrast, the SL2 environment, with high severity levels, tightens its reliability requirements at the expense of cost efficiency.

5.5.5 Phase 5. Model Formulation

This phase is a critical part and determines the success of adaptation. Figure 5.3 depicts MirrorNet as a control system consisting of a SPECTRA controller and a MirrorNet network on the plant side. To achieve the objectives, i.e., satisfice the NFRs, the controller needs to observe and analyse the plant behaviour (using models and thresholds) and decide the best action to take (e.g., topology), using the available weight vector and policy, then send the decision to the actuator. Therefore, plant behaviour needs to be accurately reflected in the model.

Based on the observabilities of the NFRs, we can determine the appropriate

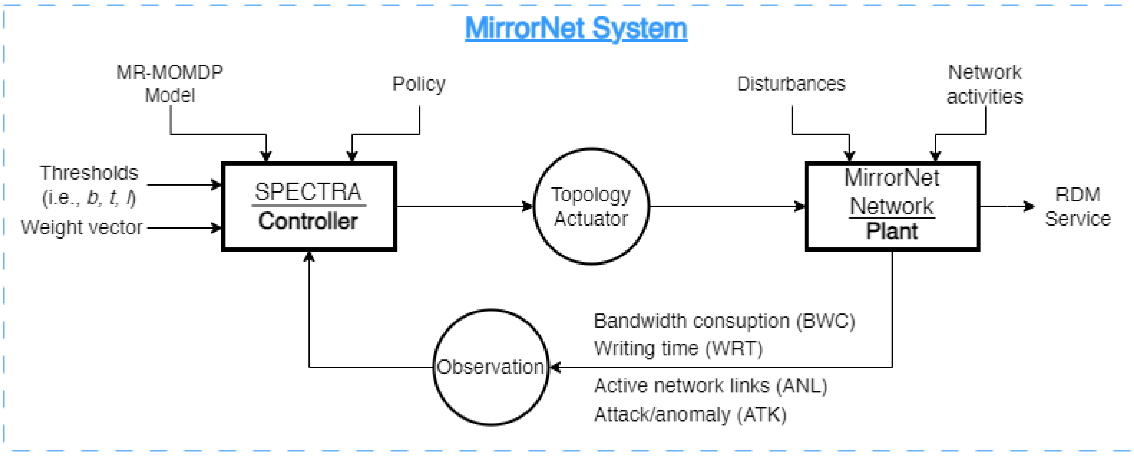


Figure 5.3: The block diagram of MirrorNet as a closed loop control system.

Markovian model to utilise. If all NFRs are fully observable, using a multi-reward MDP (MR-MDP) is recommended. However, if all NFRs are only partially observable, the problem must be modelled as an MR-POMDP. In cases where the environment presents a combination of both partially and fully observable NFRs, it is advisable to model it as an MR-MOMDP.

As shown in Table 5.2, MirrorNet exhibits mixed observability NFRs, where MC and MP can be directly observed, but MR is only partially observed through the number of active network links (ANL) and the status of network attacks (ATK). Therefore, the suggested modelling approach is MR-MOMDP. The MirrorNet environment and NFRs can be mapped to an MR-MOMDP model shown in Figure 5.4.

States

States at least represent combinations of satisfaction levels of the NFRs (i.e., MC, MP, and MR) but can also involve other variables deemed necessary to represent the states of the plant (i.e., MirrorNet network as the managed asset). In the MR-MOMDP, there are two types of states in the state space: fully observable

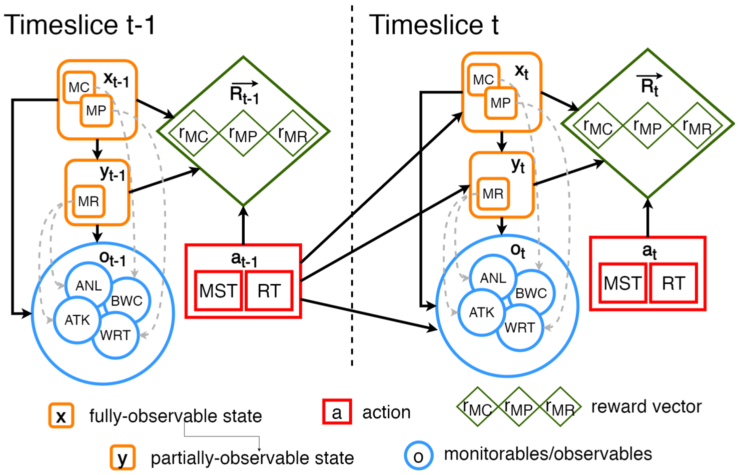


Figure 5.4: MirrorNet RDM as an MR-MOMDP problem.

Table 5.6: Fully and partially observable states of MirrorNet.

Fully observable States (X)		
State	$NFR_1 = MC$	$NFR_2 = MP$
x1	TRUE	TRUE
x2	TRUE	FALSE
x3	FALSE	TRUE
x4	FALSE	FALSE
Partially observable States (Y)		
State	$NFR_3 = MR$	
y1	TRUE	
y2	FALSE	

states and partially observable states. Regarding MirrorNet, fully observable states represent combinations of satisfaction values for MC and MP, whereas partially observable states represent combinations of satisfaction values for MR. Consequently, four fully observable states $X = \{x1, x2, x3, x4\}$ and two partially observable states $Y = \{y1, y2\}$ have been identified and presented in Table 5.6.

Actions

An action represents the agent's decision or choice in each state. In the case of MirrorNet, two adaptive actions are considered for each state: selecting the most suitable topology between Minimum Spanning Tree (MST) and Redundant Topology (RT). Based on the current state, the agent chooses the appropriate topology to

Table 5.7: NFR satisfaction probabilities.

Action	MC_{t-1}	MP_{t-1}	MR_{t-1}	P($MC_t = T$)	P($MC_t = F$)	P($MP_t = T$)	P($MP_t = F$)	P($MR_t = T$)	P($MR_t = F$)
MST	Any	Any	Any	0.9002	0.0998	0.9697	0.0303	0.3725	0.6275
RT	Any	Any	Any	0.7045	0.2955	0.8788	0.1212	0.7222	0.2778

Table 5.8: Transition probabilities to fully observable states (Tx)

Action	x_{t-1}	y_{t-1}	$x1_t$	$x2_t$	$x3_t$	$x4_t$
MST	Any	Any	0.8729	0.0273	0.0968	0.0030
RT	Any	Any	0.6191	0.0854	0.2596	0.0358

x = Fully observable state, y = Partially observable state

support the satisfaction of the NFRs in the RDM system.

Transition Functions

Any action (i.e., topology) chosen in each state will have an impact on the environment and transition the environment from the current state to a new state with some probabilities. To capture these transitions, we decompose the transition functions T_x and T_y into marginal conditional probabilities of the NFRs, specified in Table 5.7, as follows:

$$\begin{aligned}
T((x,y),a,(x',y')) &= T_x(x,y,a,x')T_y(x,y,a,y') \\
T_x(x,y,a,x') &= P(MC'|x,y,a)P(MP'|x,y,a) \\
T_y(x,y,a,y') &= P(MR'|x,y,a)
\end{aligned}$$

Table 5.8 represents the complete transition function T_x, which specifies the transition probabilities from the current state (x, y) to the next fully observable state $x' \in X$. On the other hand, Table 5.9 represents T_y, which specifies the transitions from the state (x, y) to the next partially observable state $y' \in Y$ based on the selected topology in MirrorNet.

Table 5.9: Transition probabilities to partially observable states (Ty)

Action	x_{t-1}	y_{t-1}	$y1_t$	$y2_t$
MST	Any	Any	0.3725	0.6275
RT	Any	Any	0.7222	0.2778

Rewards

In SAS's decision-making process, it is crucial to consider the satisfaction priorities of individual NFRs. SPECTRA uses vector rewards to model these priorities and considers higher reward value as higher priority/desirability when making adaptation decisions. The reward value may also represent the utility of a particular objective-action pair.

A vector reward gives different values to each objective, as opposed to a scalar reward, which gives a single value to each action. This approach provides a more precise understanding of how a particular action can impact multiple NFRs. By simultaneously evaluating several factors, a vector reward helps to make more informed decisions by capturing the relative importance and trade-offs between objectives. It enables decision-makers to optimise choices based on multiple competing objectives and more accurately represents the decision-making process.

MirrorNet aims to satisfice three NFRs, and to achieve this, each state is linked to a three-dimensional reward vector function specified as $R(s, a) = [r_{MC}, r_{MP}, r_{MR}]$. Each element of the reward vector indicates the priority of the corresponding NFR in each state compared to the other NFRs along their respective axes. The values of the reward vector are based on the elicited priorities obtained during Phase 3 - Priority Elicitation and might follow utility theories [201]. During this stage, we establish local weights as state-specific reward vectors, while the global weight vector, generated by OLSAR-Perseus in Phase 6 or determined beforehand in Phase 3, reflects the stakeholders' overall preferences.

The complete reward values are presented in Table 5.10, serving as the reward function in our model. Each entry of Table 5.10 indicates the preference over the execution of an action according to the value of NFRs in each state. We can observe that MST is preferable when all NFRs are satisfied or when MC is unsatisfied, whereas RT is preferable when MR is most likely unsatisfied. This reward function captures the cumulative impact of the NFRs on the overall reward in MirrorNet.

Table 5.10: Reward vector values for NFRs in MirrorNet.

State	Action	Reward vector values		
		r_{MC}	r_{MP}	r_{MR}
x1:y1	MST	110	95	95
x1:y2	MST	60	60	10
x2:y1	MST	95	40	80
x2:y2	MST	65	10	10
x3:y1	MST	40	80	80
x3:y2	MST	10	60	10
x3:y1	MST	35	35	65
x3:y2	MST	3	3	3
x1:y1	RT	80	95	90
x1:y2	RT	85	83	20
x2:y1	RT	60	10	60
x2:y2	RT	80	18	20
x3:y1	RT	10	66	68
x3:y2	RT	40	75	40
x3:y1	RT	11	10	26
x3:y2	RT	10	10	10

Table 5.11: Observation probabilities of MirrorNet.

MON1 Bandwidth consumption (BWC)								
	P(MC=T|BWC)				P(MC=F|BWC)			
Action	BWC $\leq$ b		BWC b		BWC $\leq$ b		BWC b	
MST	1		0		0		1	
RT	1		0		0		1	
MON2 Writing time (WRT)								
	P(MP=T|WRT)				P(MP=F|WRT)			
Action	WRT $\leq$ t		WRT t		WRT $\leq$ t		WRT t	
MST	1		0		0		1	
RT	1		0		0		1	
MON3 Active network links (ANL)								
	P(MR=T|ANL)				P(MR=F|ANL)			
Action	ANL l		ANL $\geq$ l		ANL l		ANL $\geq$ l	
MST	0.3		0.7		0.7		0.3	
RT	0.15		0.85		0.85		0.15	
MON4 Anomaly/attack status (ATK)								
	P(MR=T|ATK)				P(MR=F|ATK)			
Env:	SL1		SL2		SL1		SL2	
Action	ATK=True	ATK=False	ATK=True	ATK=False	ATK=True	ATK=False	ATK=True	ATK=False
MST	0.4	0.6	0.15	0.85	0.6	0.4	0.85	0.15
RT	0.4	0.6	0.15	0.85	0.6	0.4	0.85	0.15

Observations

Observations represent combinations of monitorable values used to determine the current state. For fully observable states, the monitorables directly determine the state. However, for partially observable states, the monitorables are used to probe the state value based on probabilistic values obtained from observations validated by domain experts.

MirrorNet incorporates four monitorables (i.e., BWC, WRT, ANL, and ATK) associated with each NFR, as outlined in Phase 2 of the MR-MOMDP and listed in Table 5.2. The observation probabilities for these monitorables in MirrorNet are presented in Table 5.11. BWC, WRT, and ANL can take on two possible values: violating or not violating the threshold, while ATK indicates the boolean status of anomalies or attacks in the network. Threshold values for BWC, WRT, and ANL (i.e., b, t, l) are given in Table 5.13. With the inclusion of these four monitorables and two possible values for each, our MR-MOMDP model for MirrorNet incorporates a total of sixteen observations $o \in \Omega \therefore |\Omega| = 16$.

5.5.6 Phase 6. Policy Planning

Given the model, our next step is to solve the problem and obtain an optimal set of policies. Policy planning can be conducted offline or online using a specific solver. If the global weight vector can be determined in Phase 3, the policy can be generated using an appropriate single-objective solver through scalarisation.

This study employs OLSAR and Perseus for offline planning, assuming that the global weight vector is not determined in Phase 3. As shown in Figure 5.5, OLSAR provides us with a set of optimal weight vectors and their corresponding vector values

Table 5.12: Selected optimal weight vectors.

Framework	Environment	
	Weight vector for SL1	Weight vector for SL2
MR-MOMDP	[0.8441, 0. , 0.1559]	[6.7026e-01, 3.5220e-06, 3.2974e-01]
MR-POMDP	[0.8387, 0. , 0.1613]	[6.2697e-01, 1.7295e-09, 3.7303e-01]

and policy. Various methods, such as knee point detection [203], can be utilised to select a values vector that aligns with the preferences. However, this study defines the minimum CCS values as the threshold to choose an optimal weight vector and its corresponding policy for each value point that exceeds the threshold. The chosen weight vector and policy are among the inputs the controller uses (i.e., managing system), as shown in Figure 5.3. Table 5.12 shows the optimal weight vectors chosen for different frameworks and environments during our evaluation.

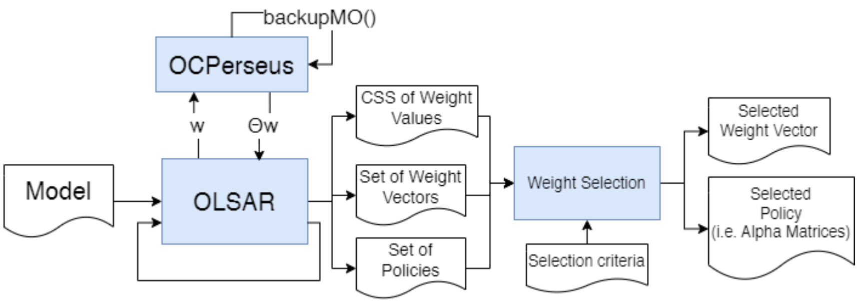


Figure 5.5: Policy planning with OLSAR & OCPerseus.

5.6 Evaluations

We evaluate the feasibility of our proposed SPECTRA and MR-MOMDP frameworks by comparing the policy planning of MirrorNet using MR-MOMDP with baseline approaches. Specifically, we aim to address the following research questions:

(**RQ5.1):** How can SPECTRA better optimise the planning process and produce better policy?

(**RQ5.2):** To what extent does SPECTRA produce better policies and effectively satisfy NFRs with tradeoffs aligning with stakeholders' preferences?

Through this evaluation, we aim to assess the effectiveness of our proposed SPECTRA framework and determine the superiority of MR-MOMDP over the baseline approaches in terms of planning time and policy quality for MirrorNet.

5.6.1 Experiment Setup

Environment Model and Scenario

To analyse the effectiveness of the policies, each approach is tested in two scenarios: the *Non-detrimental Scenario*, where all topologies operate normally without any disruptions, and the *Detrimental Scenario*, where system disruptions occur. To introduce variability in our experiments, we consider two environmental models representing the Low-Moderate Severity Level (SL1) and High Severity Level (SL2) in the Detrimental Scenario. These environments are modelled using different observation functions in MON4, as shown in Table 5.11.

Architecture

Our experiments were conducted using a modified Python version of RDMSim [188], a discrete event simulator of a RDM environment, which we put as the managed asset at the bottom layer as shown in Figure 5.6. We modified the original code to ensure reproducibility through a random seed function. We conducted simulations and experiments on a laptop with an Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz and 8 GB RAM running Windows 10 Pro.

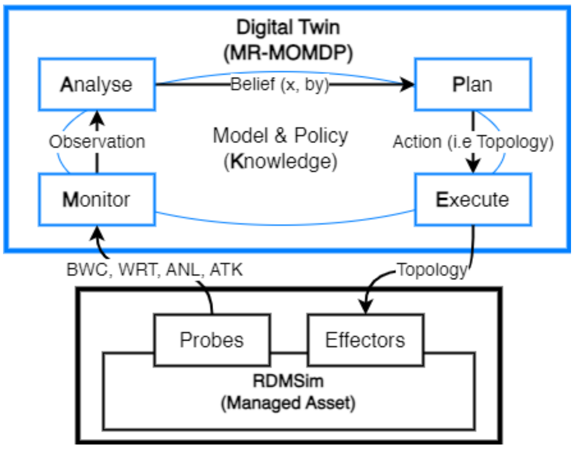


Figure 5.6: Digital twins architecture used in the evaluation with RDMSim and MR-MOMDP.

Table 5.13: RDMSim configuration

Configuration item	Value
Random seed	0
Time steps	200
Mirrors	25
MST Active Links %	[8, 58]
RT Active Links %	[30, 65]
Single bandwidth consumption	[20, 30]
Single writing time	[10, 20]
BWC max. threshold b	4050
WRT max. threshold t	3000
ANL min. threshold l	90

The digital twin at the upper layer is the managing system that acts as a decision-making agent with MAPE-K components. It employs MR-MOMDP or other baseline approaches in the Analysis & Planning component. Based on observations made by the Monitoring component, the Analysis component determines its belief about the current state. The planning component uses this belief state to select the appropriate topology based on the policy provided by the Knowledge component. The Execute component then communicates the decision to the Effector component in RDMSim, which activates the topology according to the instructions.

We configured RDMSim to simulate MirrorNet, which consists of 25 mirrors with 300 physical links used for data transfer between these mirrors, with the settings specified in Table 5.13.

5.6.2 Evaluation Metrics

We used the following metrics to analyse the experimental results:

1. Planning time: the execution time required to generate a CCS of weight vectors and a set of policies.

2. Expected total reward: the accumulated expected reward obtained by executing an action in each step, in which the belief state determines the value for each step.

3. Size of CCS solutions: the number of non-dominated weight vectors successfully found.

4. Topology proportions: the proportion of selecting MST topology relative to RT topology in a simulation round.

5. Target satisfaction: measures how closely the execution results meet the target.

6. Constraints violation: identifies whether the execution results violate the given constraints.

5.6.3 Baselines

MR-POMDP

For a fair comparison, we selected the implementation of MR-POMDP that also utilises OLSAR-Perseus [189, 29] to investigate the effects of modelling multi-objective mixed observability problems using MR-MOMDP (a specific model for mixed observability) and MR-POMDP designed for multi-objective partial observability.

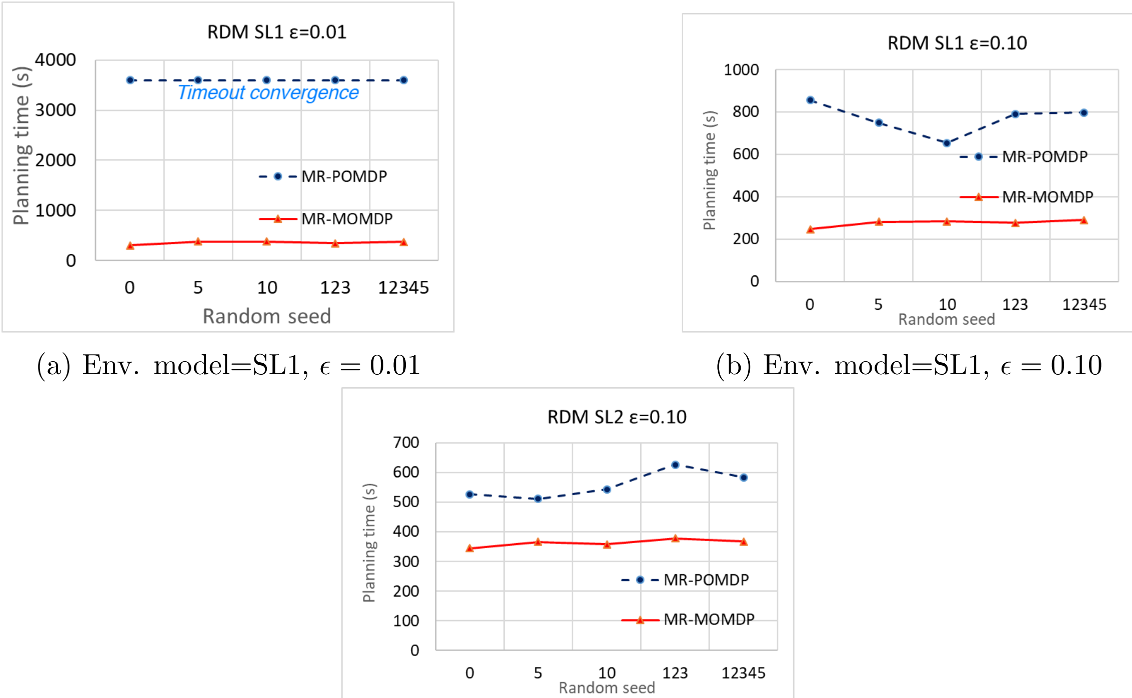
Rule-Based Adaptation

. RDMSim [188] is equipped with a default rule-based adaptation mechanism that selects the MST topology if there is a violation of the BWC threshold and WRT threshold and chooses the RT topology if the ANL threshold is violated. We use this as an additional baseline to compare various adaptation strategies for satisficing NFRs.

5.6.4 (RQ5.1) Optimal Policy Planning

The SPECTRA method enhances the management of tradeoffs in uncertain situations for SAS. This is achieved through three key features: 1) considering diverse observability to determine the most suitable MDP model, 2) priority elicitation to guide policy decision-making during runtime, and 3) representing the priorities of each NFR through vector reward to facilitate well-informed decision-making.

Through SPECTRA, we can model the decision-making of MirrorNet using MR-MOMDP, which can be compared with baselines. As previously explained, we implemented MR-MOMDP with OLSAR and Perseus to formulate an optimal policy configuration for MirrorNet, using a convergence threshold of $\eta = 0.01$, a maximum allowed error of $\epsilon = 0.01$, and a convergence timeout of 3600 seconds. We compared the planning time required by MR-MOMDP and MR-POMDP in two environments with different values of ϵ. In the default setting, as shown in Figure 5.7a, with $\epsilon = 0.01$, MR-POMDP is intractable and always reaches the timeout convergence, while MR-MOMDP completes within the range of 300-372 seconds. When the maximum allowed error is relaxed to $\epsilon = 0.10$, MR-MOMDP still finishes faster than MR-POMDP, taking 247-291 seconds in the SL1 environment (Figure 5.7b) and 345-379 seconds in the SL2 environment (Figure 5.7c), while MR-POMDP requires



(a) Env. model=SL1, $\epsilon = 0.01$

(b) Env. model=SL1, $\epsilon = 0.10$

(c) Env. model=SL2, $\epsilon = 0.10$

Figure 5.7: Planning time of MR-POMDP and MR-MOMDP in different settings

654-856 seconds in the SL1 environment and 511-627 seconds in the SL2 environment. It can also be observed that different random seeds yield different planning times.

We observed that the backup time required by MR-MOMDP is longer than MR-POMDP due to the additional loop caused by the decomposition of states into fully and partially observable states. This decomposition increases the number of alpha-matrices and computational operations needed in MR-MOMDP. However, we suspect that the smaller belief space maintained by MR-MOMDP reduces the number of corner weights that need to be tested individually. On the other hand, MR-POMDP with a larger belief space has a higher likelihood of finding more corner weights, which leads to a longer convergence time for MR-POMDP.

Figure 5.8 shows the size of CCS found by MR-POMDP and MR-MOMDP using $\epsilon = 0.01$, where MR-POMDP yields 174 points in the SL1 environment and

Table 5.14: CCS size of MR-MOMDP and MR-POMDP in different environments and settings

<table>
<tr><th colspan="6">CCS Size</th></tr>
<tr><th colspan="6">Part 1. Seed=0, Epsilon=0.01</th></tr>
<tr><td></td><th colspan="3">Env: SL1</th><th colspan="2">Env: SL2</th></tr>
<tr><td>MR-MOMDP</td><td colspan="3">18</td><td colspan="2">34</td></tr>
<tr><td>MR-POMDP</td><td colspan="3">174</td><td colspan="2">164</td></tr>
<tr><th colspan="6">Part 2. Env: SL1, Epsilon=0.10</th></tr>
<tr><th>Seed</th><th>0</th><th>5</th><th>10</th><th>123</th><th>12345</th></tr>
<tr><td>MR-MOMDP</td><td>8</td><td>8</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>MR-POMDP</td><td>93</td><td>87</td><td>81</td><td>89</td><td>98</td></tr>
<tr><th colspan="6">Part 3. Env: SL2, Epsilon=0.10</th></tr>
<tr><th>Seed</th><th>0</th><th>5</th><th>10</th><th>123</th><th>12345</th></tr>
<tr><td>MR-MOMDP</td><td>12</td><td>12</td><td>12</td><td>12</td><td>13</td></tr>
<tr><td>MR-POMDP</td><td>74</td><td>70</td><td>77</td><td>85</td><td>78</td></tr>
</table>

164 points in the SL2 environment, while MR-MOMDP only yields 18 points and 34 points, respectively. However, when the maximum allowable error is relaxed to $\epsilon = 0.10$, we can reduce the size of the CCS in MR-POMDP to approximately half, as shown in Table 5.14.

As we need to decide on a single final weight vector along with its corresponding set of policies, having a larger CCS size provides more alternatives. However, it also requires more time and effort to determine the choice, especially if the selection is made manually by a human decision-maker. Nevertheless, having a reliable autonomous component to select the best final weight vector will benefit the planning process.

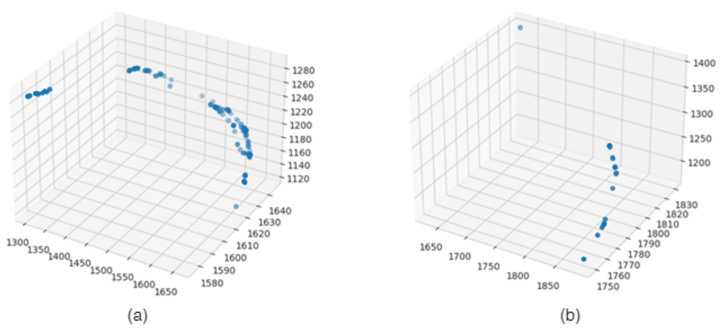


Figure 5.8: CCS of (a) MR-POMDP (164 points) and (b) MR-MOMDP (34 points) under SL2 context.

5.6.5 (RQ5.2) Policy Quality and NFR Satisfaction

We answer RQ5.2 by evaluating the policies produced by MR-MOMDP and MR-POMDP using metrics: expected total reward, the proportion of topology used, target satisfactions, and constraint violations, which we discuss below.

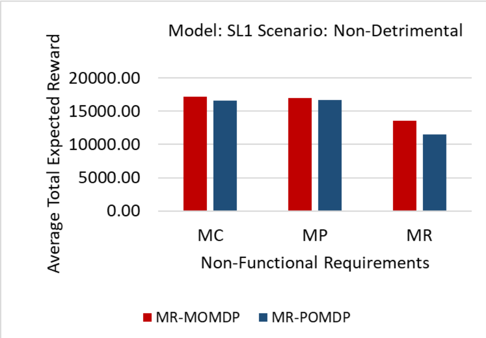
Expected Total Reward

The expected total reward represents the utility value of a system because the agent's goal is to maximise the expected total reward through a sequence of actions while accounting for the uncertainty of the system's state. Therefore, a set of policies that yields a higher expected total reward can be considered a better set of policies.

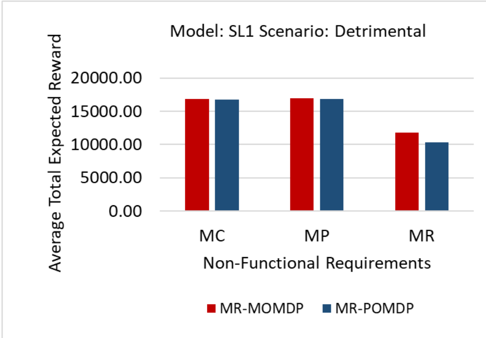
We compared the average expected total reward of MR-MOMDP and MR-POMDP for each NFR in SL1 and SL2 environments under both non-detrimental and detrimental conditions, using different random seeds mentioned in Table 5.14 and Figure 5.7.

Figure 5.9 shows that under non-detrimental conditions, all NFRs had higher average expected total rewards in MR-MOMDP and MR-POMDP compared to detrimental conditions. This outcome is not surprising since NFRs are easier to satisfice in normal conditions than during anomalies. In non-detrimental conditions, SL2 provided higher rewards than SL1, while in detrimental conditions, SL1 performed better than SL2. The reason behind this is the probability values in the observation function, where SL2 has higher probability values of P(MR=T|ATK=F) and lower values of P(MR=T|ATK=T).

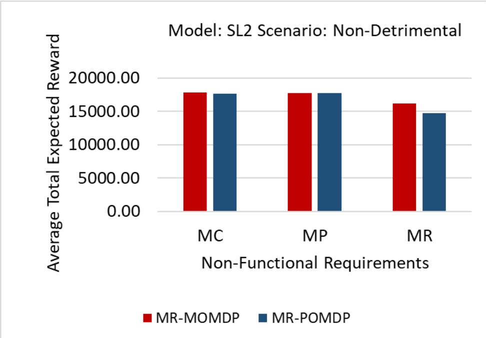
Overall, MR-MOMDP yielded higher average values for all NFRs compared



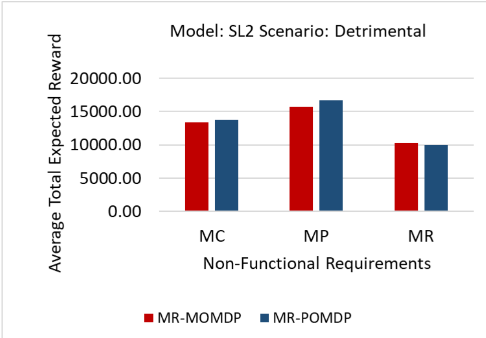
(a) Env. model = SL1, Scenario = Non-Detrimental



(b) Env. model = SL1, Scenario = Detrimental



(c) Env. model = SL2, Scenario = Non-Detrimental

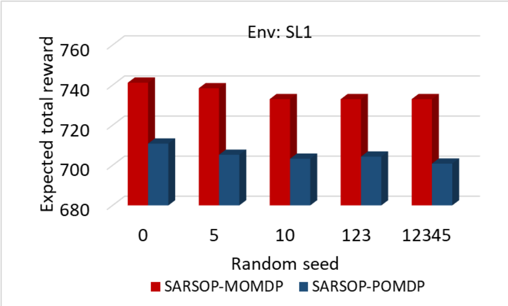


(d) Env. model = SL2, Scenario = Detrimental

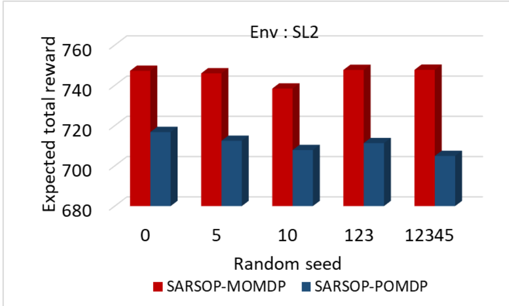
Figure 5.9: Expected total reward of MR-MOMDP and MR-POMDP in different environment models.

to MR-POMDP. However, in SL2-Detrimental, MR-POMDP outperformed MR-MOMDP in MC and MP, as MR-MOMDP sacrificed both NFRs for higher MR, which is a preferred outcome.

To ensure the reward difference was not due to the weight vector choice, we evaluated the solutions using the policy evaluator tool provided by APPL [128]. We compared the scalarized versions of MR-MOMDP and MR-POMDP with a weight vector of [0.333, 0.333, 0.333], solved using SARSOP [128]. Figure 5.10 shows that the scalarized MR-POMDP (SARSOP-MOMDP) consistently yielded higher expected rewards than SARSOP-POMDP across various configurations. This is because MOMDP reduces uncertainty through a smaller belief space, resulting in higher belief values that impact optimal rewards.



(a) Env. model=SL1



(b) Env. model=SL2

Figure 5.10: Expected Total Reward of Scalarised MR-MOMDP and MR-POMDP using SARSOP

Topology Proportion

As highlighted in Section 5.5, the utilisation of the MST topology offers the advantage of lower operational expenses, although it may not provide the same level of reliability as the RT topology. The MST topology can be a suitable option when the risk of disruptions is low, such as in non-detrimental conditions. However, it may not be ideal in detrimental situations where reliability is crucial.

In Figure 5.11, the usage proportions of the MST and RT topologies are depicted for the SL1 and SL2 environments, respectively. In non-detrimental conditions, MR-MOMDP, MR-POMDP, and the Rule-based adaptation demonstrate similar proportions in terms of utilising the two topologies. However, in detrimental conditions, there are notable differences.

The rule-based adaptation only marginally increases the allocation of RT by approximately 9-11% in detrimental conditions. On the other hand, both MR-MOMDP and MR-POMDP significantly increase the portion of RT. In the SL1 environment, they allocate RT by approximately 20%. In the more restrictive SL2 environment, MR-MOMDP and MR-POMDP significantly increase the allocation of RT by about 70%. This shift towards RT usage is driven by the need to adhere

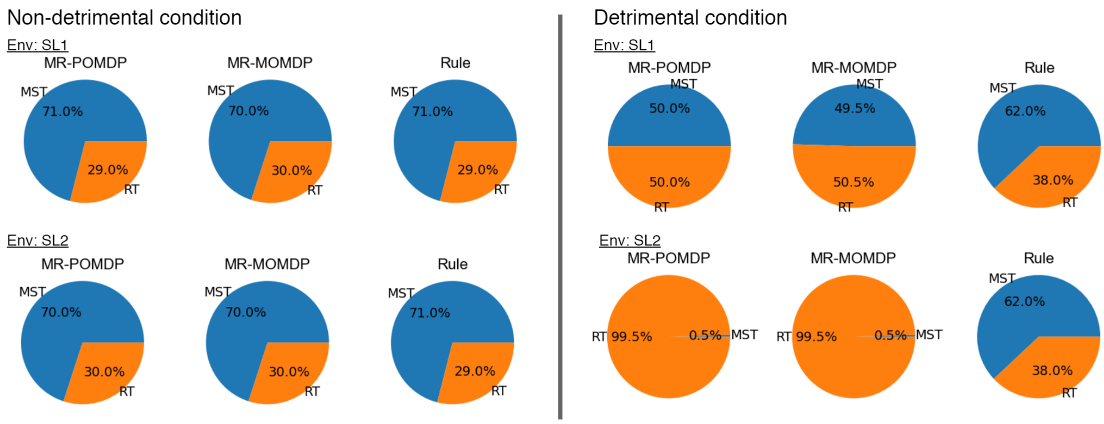


Figure 5.11: Proportions of topology under different conditions.

to stricter MR constraints in SL2-Detrimental compared to SL1-Detrimental. It is important to highlight that the higher reliance on RT in SL2-Detrimental leads to increased operational costs compared to SL1-Detrimental.

Target Satisfaction

Based on the specified targets in Table 5.3, the objective is to minimise the average values of WRT (MP) and BWC (MC) while maximising the average value of ANL (MR). Figure 5.12 illustrates that under normal or non-detrimental conditions, MR-MOMDP, MR-POMDP, and the Rule-based adaptation perform equally well in achieving all the targets. However, in detrimental conditions, a notable difference arises. The Rule-based adaptation yields a lower WRT and BWC average than MR-MOMDP and MR-POMDP. This discrepancy can be attributed to the rule-based adaptation relying more heavily on the MST topology than MR-MOMDP and MR-POMDP. Consequently, when detrimental conditions occur, disruptions in the MST topology reduce the number of active links, resulting in decreased values of both BWC and WRT.

On the contrary, MR-MOMDP and MR-POMDP adapt their strategy by increas-

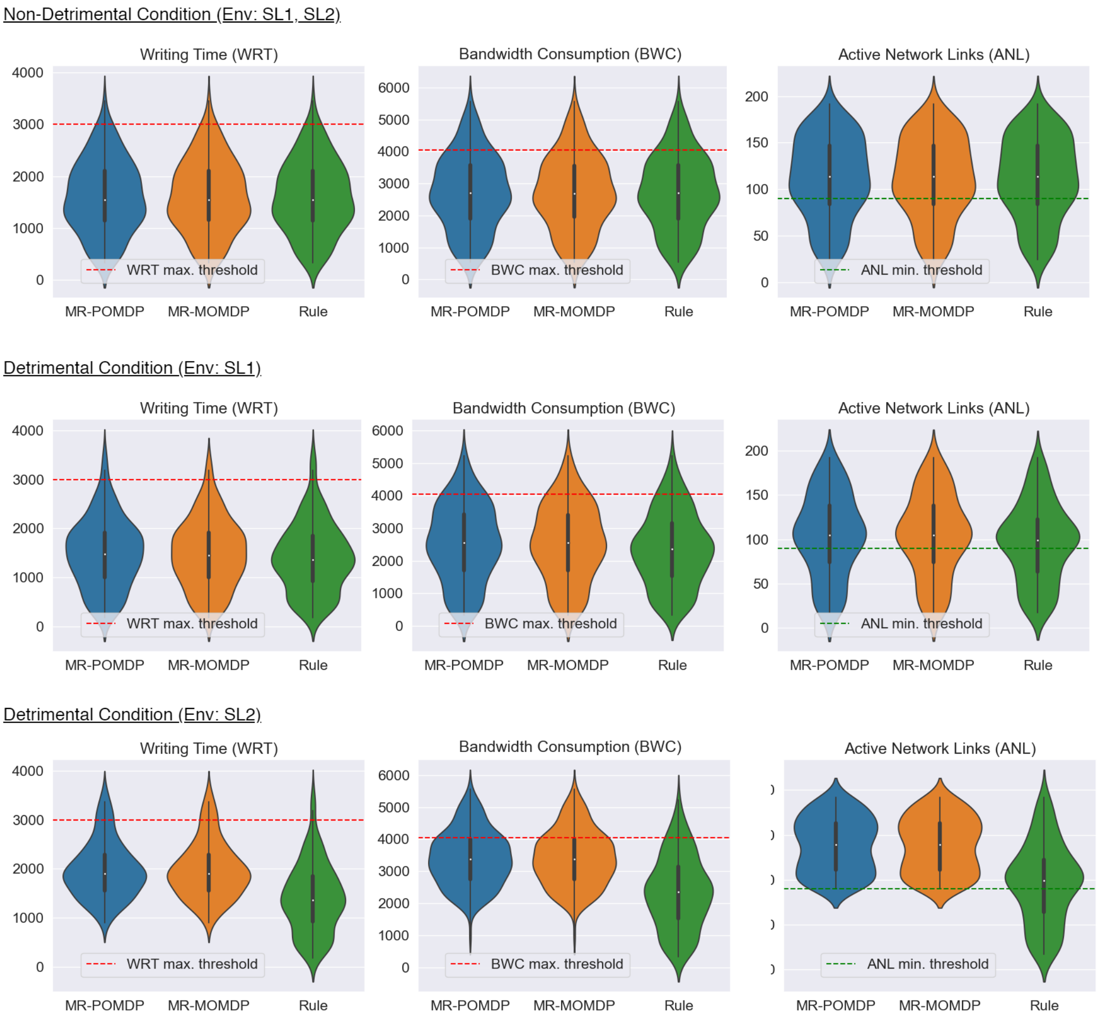


Figure 5.12: Target satisfaction under various conditions.

ing the allocation of the RT topology to maintain system reliability. This strategic adjustment allows them to equally satisfy all the targets while outperforming the Rule-based adaptation in satisficing the ANL objective.

To summarize, under detrimental conditions, the Rule-based adaptation exhibits lower values of WRT and BWC due to its reliance on the MST topology, which is more susceptible to disruptions. In contrast, MR-MOMDP and MR-POMDP respond by prioritizing the RT topology to enhance system reliability, resulting in better performance across all targets and excelling in satisficing the MR requirements.

Constraint Violation

Regarding constraint violation, as shown in Figure 5.13, none of the approaches, including MR-POMDP, MR-MOMDP, and rule-based adaptation, violate any constraints under non-detrimental conditions. However, when subjected to detrimental conditions, the rule-based adaptation violates the MR constraints in SL1 and SL2, indicating its inability to handle detrimental conditions effectively. On the other hand, both MR-POMDP and MR-MOMDP still manage to avoid violating any constraints. Additionally, during the evaluation, we specifically observed the number of threshold

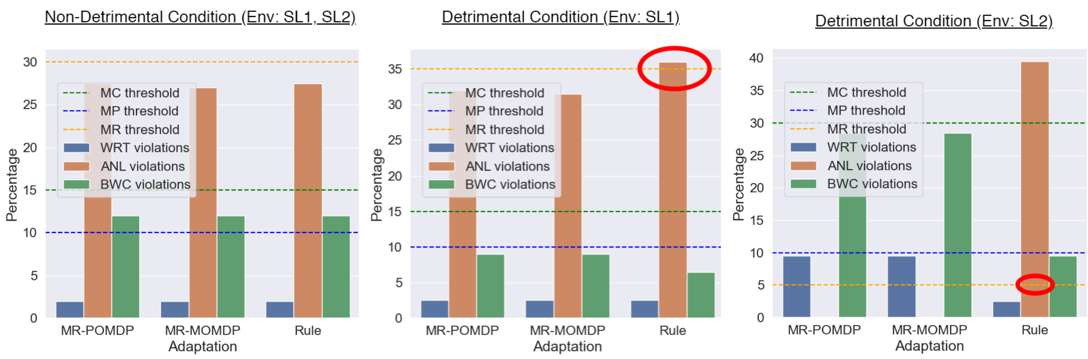


Figure 5.13: Constraint violation under various conditions.

violations between MR-MOMDP and MR-POMDP across different random seeds. The aim was to assess the performance of the two approaches in maintaining the predefined thresholds for the NFRs. Notably, our findings indicate that the number of violations is comparable between MR-MOMDP and MR-POMDP. This implies that the two approaches exhibit similar capabilities in effectively maintaining the set thresholds for the NFRs.

Based on these findings, we can conclude that the default RDMSim rule-based adaptation cannot effectively satisfy the NFRs in this experiment. In contrast, MR-POMDP and MR-MOMDP exhibit better performance by successfully adhering to the MR constraints, even under detrimental conditions. However, one can improve the rule-based adaptation by adding more rules to address different conditions, which is often impractical.

5.7 Software Implementation

We implemented MR-MOMDP in a Python framework called XPOMDPy, which stands for Extended POMDPy. For the purpose of reproducibility, the framework is available online at [204]. XPOMDPy extends the current POMDPy framework [190] by providing the following additional features:

1. **Perseus solver for single-objective POMDP and MOMDP, as well as OLSAR-Perseus for multi-objective MR-POMDP and MR-MOMDP.** Our implementation of Perseus is a Python translation of Java implementation by [205]. We also use OLS implementation by [206] in our OLSAR implementation.
2. **Model parsing capabilities from POMDP and PomdpX file formats.** This feature simplifies the process of solving and generating new model objects by easily reading and interpreting the model specifications.
3. **Policy writer that supports the PolicyX file format for both single-objective (i.e., alpha vector) and multi-objective (i.e., alpha matrix) policies.** This enables storing and retrieving policies in a structured format, facilitating their use in decision-making processes.

5.8 Threats to Validity

Threats to external validity are associated with the generalisability of results and the replicability of experiments. Our testing was conducted using RDMSim simulator

[188], which has also been used in [29, 126]. We present two models and different environmental scenario conditions with the initial configuration provided by domain experts. More complex environments and problems will be more challenging to model and may yield different test results. Another issue is the computational cost of both MR-MOMDP and MR-POMDP, which can become intractable in their worst-case scenarios when the number of states used is too large, even though MR-MOMDP requires a shorter computation time than MR-POMDP. Our approach adopts [29, 124], which uses states as representations of combinations of two values for the satisfaction levels of NFRs (i.e., True and False). For example, 2 NFRs will result in 4 states, 3 NFRs will yield 8 states, and so on. Therefore, limiting the number of NFRs used to drive self-adaptation is recommended when implementing our proposal. To strive for replicability, we provided a detailed experiment setup and configuration used in the simulator. Our implementation code, model, and simulator are also available online for the public.

Threats to internal validity can be associated with randomness and settings in the implementation. We observe that the randomness of the environment affects the test results. Changes and differences in values for each component in the model will result in different policies. We employed multiple random seeds to ensure consistent experimental results and represent diversity in the test outcomes. To minimise bias in settings and implementation, we presented test results with different constraint settings according to the environment. We compared MR-MOMDP with MR-POMDP using the same solver with comparable weight vectors, as well as through scalarization using the same weight vector.

Threats to construct validity are related to the suitability of the metrics used. To anticipate this, we utilised various metrics in the evaluation to address our research questions, including planning time, expected total reward, CCS size, target satisfaction, and constraint violation.

5.9 Positioning of The Study

We have discussed relevant literature that aligns closely with our proposal in Section 2.4.3. Our focus is solely on those works that employ the Markov model within the area of decision-making for self-adaptation. A comparison of our proposal with some of the most relevant previous work is summarised in Table 5.15.

Table 5.15: Comparison of our proposal with existing approaches

Approach	# of NFRs	State space represents NFR satisfaction information?	Observability			Reward
			Full	Partial	Mixed	
Hybrid planning [116]	Single	No	✓			Scalar
CSSC-MDP [113]	Multiple	Yes	✓			Scalar
ADAM [114]	Multiple	Yes	✓			Scalar
MDP-LSP with HTN [112]	Multiple	Yes	✓			Scalar
Wang et al. [115]	Single	Yes		✓		Scalar
RE-STORM-ARRoW [125]	Multiple	Yes		✓		Scalar
MR-POMDP++ [29]	Multiple	Yes		✓		Vector
SPECTRA (our proposal)	Multiple	Yes	✓	✓	✓	Vector

Our proposal, SPECTRA, builds upon RE-STORM [124] and Pri-AwaRE [126] by incorporating a state space model to represent the satisfaction level of NFRs. We recognise the significance of using MDP and POMDP in environments with uniform observability. However, unlike existing approaches, our work takes a step further by incorporating MOMDP to handle decision-making in mixed observability environments, an under-explored area in Software Engineering and Requirements Engineering in particular.

Echoing the importance of explicitly defining the significance of each NFR through vector rewards, our work goes beyond MR-POMDP++ by considering the mixed observability of using MOMDP and contributing to a multi-reward version of MOMDP (i.e., MR-MOMDP, described in section 5.3) which can be applied in various domains. The SPECTRA we propose covers the multi-reward/vector reward MDP, MR-POMDP, and MR-MOMDP and serves as a guideline for designing priority-aware decision-making models of SAS to satisfy NFRs tradeoffs.

5.10 Conclusion

We have proposed SPECTRA, a framework that can be used as a guide in designing decision-making tradeoffs for satisficing multiple NFRs in SAS using various Markov models according to the observability of the NFRs involved. Through the Mirror-Net scenario, we have shown that MR-MOMDP is more suitable for contexts with NFRs showing mixed observability. The evaluation shows that MR-MOMDP can shorten planning time, produce higher expected values total rewards, and better satisfy NFRs. These findings encourage further development and exploration in other domains with mixed observability.

Chapter 6

Exploring SPECTRA within the Context of HitLCPS

In Chapter 5, we demonstrate and evaluate SPECTRA and compare MR-MOMDP with MR-POMDP using the remote data mirroring scenario. This chapter illustrates the application of SPECTRA in the context of HitLCPS through a case study of a credit card payment system facilitated by MultiMAUS, a simulator designed for online credit card transactions with multi-modal authentication capabilities, which we implement using the same digital twin architecture as in Chapter 5. Within this system, humans act as both genuine and fraudulent service customers.

We introduce a new heuristic authentication approach for MultiMAuS, dubbed "Dynamic", and showcase the utilisation of the SPECTRA framework for specifying NFRs and facilitating self-adaptive authentication. Experiment results indicate that SPECTRA surpasses built-in methods by offering superior trade-offs, notably enhancing the performance of Dynamic, particularly in scenarios characterised by a higher frequency of fraudulent activities.

6.1 Introduction

In our previous instantiation of SPECTRA, we used the remote data mirroring scenario, where state spaces solely represented the satisfaction level of NFRs. However, decisions often need to be made while also considering other states of the environment. In HitLCPS, humans can be part of the observed environment. In many HitLCPS applications, the decision maker (i.e., controller) cannot solely focus on the satisfaction level of NFRs but needs to consider both external and internal human states to achieve an optimal tradeoff.

HitLCPS differs from other CPS in that it involves humans in the feedback loop, serving as operators/decision-makers, service consumers, and service providers. When humans are part of the entities being monitored, it adds a layer of complexity and observability.

Human behaviour is complex and can be influenced by a variety of factors, including emotions, context, and social interactions [207, 27]. It requires contextual information to understand human behaviour [208, 209]. However, contextual information is often insufficiently available or difficult to interpret.

When monitoring humans in real-time, there are temporal dynamics to consider, as observations may change quickly [210]. This makes obtaining a complete and accurate understanding of their behaviour challenging. Therefore, in many HitLCPS applications, an approximate approach is commonly adopted since human internal states are considered partially observable [211].

SPECTRA, based on a probabilistic approach used in decision-making under uncertainty, provides flexibility by allowing an additional factor to be represented in the state space within SPECTRA apart from NFRs satisfaction level. This chapter demonstrates how SPECTRA can handle and provide better tradeoffs in such situations, commonly encountered in HitLCPS using the scenario of a credit card payment system.

6.2 Case Study: PayNet, a Credit Card Payment System

Consider PayNet, a credit card payment system with three main components: merchants, customers, and a payment processing platform. PayNet is an instance of HitLCPS because it incorporates humans, autonomous agents, and other physical components (e.g., merchants). It explicitly involves humans as service customers and an autonomous payment processing platform that monitors human transactions and optimises customer (i.e., human) satisfaction.

Customers and merchants complete payments via the payment processing platform. The payment processing platform earns commission from each successful genuine transaction (i.e., reward) and compensates customers for losses resulting from successful fraudulent transactions (i.e., loss).

For every successful genuine transaction, the reward is calculated as 0.3% of the transaction amount, in addition to 1 cent in Euro. This reward is earned by the online payment platform for facilitating the transaction between the cardholder and the merchant. If a fraudulent transaction goes undetected, the loss is equivalent to the negative amount of the transaction. This loss represents the financial liability incurred by the online payment platform when a fraudulent transaction goes undetected, and the funds need to be reimbursed to the genuine cardholder.

By considering both the rewards earned from genuine transactions and the losses incurred from undetected fraudulent transactions, the online payment platform can evaluate the effectiveness of different authentication mechanisms in minimising losses and maximising revenue.

6.3 SPECTRA Framework for PayNet

The following sections will illustrate how SPECTRA can be applied to the PayNet credit card payment system use case.

6.3.1 Phase 1. NFR specification

Stakeholders and designers define NFRs through three main attributes (i.e., name, scale, and metre). Initially, each NFR is named, reflecting aspects like configurability, performance, security, or reproducibility. Secondly, they specify the scale, identifying what is being measured and its unit of measurement. Lastly, they establish the metre, outlining the measurement method. Stakeholders and designers must agree on the NFRs, adjusting participants as needed, and revisiting them to adapt to market, organisational, or technological changes.

As a credit card payment system, PayNet needs to provide a secure platform with minimal fraud to minimise losses (MINF). On the other hand, PayNet also needs to maximise user convenience and satisfaction (MAXS) by avoiding unnecessary authentication. Table 6.1 illustrates how the name, scale, and metre for PayNet are defined. Fraudulent transactions and customer satisfaction levels serve as scales for MINF and MAXS, respectively, and are measured using their respective averages.

Table 6.1: Phase 1 - NFR Definitions

Name	**Scale**	**Meter**
Minimise fraudulent transactions **(MINF)**	Fraudulent transactions	Average number of fraudulent transactions
Maximise customer satisfaction **(MAXS)**	Customer satisfaction level	Average customer satisfaction level

6.3.2 Phase 2. Observability Identification

In this stage, stakeholders and designers evaluate the observability of metrics linked to NFRs. Metrics are classified as either direct or indirect observability. If a metric can be directly observed, the NFR is considered fully observable; otherwise, if the metric requires probing through different monitorable objects, the NFR is partially observable. For fully observable metrics, the metric itself serves as the monitorable, while for partially observable metrics, different objects are used as monitorables to examine the desired metric.

Table 6.2

NFR	**Metrics**	**Observability**	**Monitorables**
MINF	Average number of fraudulent transactions	Direct	SMA* of fraudulent transactions: 24 hours (**SMAF24**), and 168 hours (**SMAF168**)
MAXS	Average customer satisfaction level	Indirect	SMA* of used-card transactions: 24 hours (**SMAS24**), and 168 hours (**SMAS168**)

* SMA = Simple Moving Average

As shown in Table 6.2, we can observe the number of fraudulent transactions directly. We utilise Simple Moving Averages with rolling periods of 24 hours and 168 hours and compare them to measure the number of fraudulent transactions (SMAF24, SMAF168) and used-card transactions (SMAS24, SMAS168). MINF can be directly determined by comparing SMAF24 to SMAF168. Meanwhile, MAXS is approximated from the comparison of SMAS24 to SMAS168 following the probabilities listed in Table 6.9.

6.3.3 Phase 3. Priority Elicitation

In this stage, stakeholders can balance NFRs by creating a utility function that assigns importance to each quality attribute. This can be accomplished by developing a weight vector, incorporating different quality attribute scenarios, or using conditional logic.

> "The system must ensure that fraudulent transactions are minimised as much as possible. If it can be considered that the customer is genuine, then the system can relax its authentication to maximise customer satisfaction."

From the conditional logic above, we can interpret that PayNet generally prioritises MINF over MAXS. This is also reflected in the following phases.

6.3.4 Phase 4. Targets and Constraints

This phase is essential for setting clear objectives and boundaries that determine the system's satisfaction level. Targets and constraints are defined using monitorables rather than meters or metrics, as monitorables are directly observable and can be influenced by actions or decisions taken in each state. This allows stakeholders to communicate expectations effectively and align the system with organisational goals. However, targets and constraints may be relaxed under specific circumstances to enable the system to adapt to changing and uncertain environments while still meeting critical requirements.

As shown in Table 6.3, we use RELAX to define targets for PayNet, allowing room for adaptation. PayNet aims to keep SMAF24 as low as possible and not exceed

Table 6.3: Targets of PayNet

NFR	Target
MINF	SMAF24 SHALL NOT be higher than SMAS168 and SHALL be kept AS LOW AS POSSIBLE
MAXS	SMAS24 SHALL be higher than SMAS168 and SHALL be kept AS HIGH AS POSSIBLE

Table 6.4: Constraints of PayNet

NFR	Constraint
MINF	The percentage of conditions where SMAF24 is higher than SMAS168 < 5%
MAXS	The percentage of conditions where SMAS24 is lower than SMAS168 <= 35%

SMAF168. This is done so that if there is a spike in successful fraudulent transactions, PayNet can tighten its authentication as early as possible until it reaches the MINF target again. In Table 6.4, we define constraints to limit violations of the MINF target to less than 5%.

Meanwhile, SMAS24 is expected to consistently exceed SMAS168, to maintain a positive trend in transactions using used cards, which correlates positively with customer satisfaction. As shown in Table 6.4, constraints for MAXS are looser than MINF, allowing violations of the target to be less than or equal to 35%. This is because priority elicitation, done in Phase 3, emphasises MINF over MAXS.

6.3.5 Phase 5. Model Formulation

We can select a suitable Markovian model by considering the observability of NFRs and other required states. As shown in Table 6.2, PayNet has two NFRs with mixed observability, therefore, we formulate the problem as MR-MOMDP, shown in figure 6.1. A detailed description of figure 6.1 is discussed in the following sections.

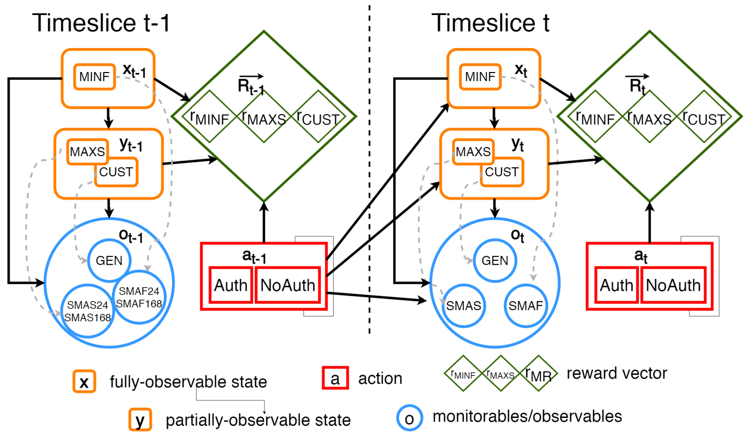


Figure 6.1: PayNet as an MR-MOMDP problem.

State space

Referring to our proposed SOA-HitLCPS discussed in Chapter 3, each node in HitL-CPS has its own function and task, which generally corresponds to its context. Context often becomes a determining factor in decision-making at runtime. Therefore, although states essentially represent the satisfaction level of each NFR, additional states may also be included to represent contexts that are necessary to consider during decision-making.

For PayNet, as shown in Table 6.5, we define two NFRs (MINF and MAXS) and include an additional variable representing customer authenticity (CUST). Customer authenticity is a crucial factor (i.e., context) in determining whether PayNet needs to request a second authentication or not.

Thus, the state space in PayNet consists of two fully observable states, $X = \{x1, x2\}$ representing the satisfaction level of MINF and four partially observable states, $Y = \{y1, y2, y3, y4\}$ representing combinations of the possible value of MAXS and customer genuineness (CUST).

Table 6.5: States in PayNet

Fully observable states (X)		
State	NFR1 = MINF	
x1	TRUE	
x2	FALSE	
Partially observable states (Y)		
State	NFR2=MAXS	CUST
y1	TRUE	TRUE
y2	TRUE	FALSE
y3	FALSE	TRUE
y4	FALSE	FALSE

Actions

In every state, the agent must make a decision represented by actions in the model. PayNet employs two actions: *Auth*, indicating that the authenticator requests a second authentication to authorise the transaction, and $NoAuth$, meaning that the transaction can proceed without a second authentication.

The action taken is considering the observations available and the expected future rewards based on the current belief about the current state, which represents the satisfaction level of NFRs and the authenticity of the customer.

Transition Functions

The action taken determines whether the next state remains the same or changes to a different state. This transition follows specific probabilities defined in the transition functions T_x and T_y. Table 6.6 represents the complete transition function T_x of PayNet, specifying the transition probabilities from the current state (x_{t-1}, y_{t-1}) to the next fully observable state $x_t \in X$. Meanwhile, Table 6.7 represents the complete transition function T_y of PayNet, specifying the transition probabilities from the current state (x_{t-1}, y_{t-1}) to the next partially observable state $y_t \in Y$.

Table 6.6: Transition function (T_x) of PayNet

Action	x_{t-1}	y_{t-1}	x_t	
			x1	x2
Auth	x1	y1	0.88	0.12
Auth	x1	y2	0.86	0.14
Auth	x1	y3	0.65	0.35
Auth	x1	y4	0.61	0.39
Auth	x2	y1	0.58	0.42
Auth	x2	y2	0.56	0.44
Auth	x2	y3	0.54	0.46
Auth	x2	y4	0.51	0.49
NoAuth	x1	y1	0.75	0.25
NoAuth	x1	y2	0.74	0.26
NoAuth	x1	y3	0.6	0.4
NoAuth	x1	y4	0.55	0.45
NoAuth	x2	y1	0.45	0.55
NoAuth	x2	y2	0.25	0.75
NoAuth	x2	y3	0.3	0.7
NoAuth	x2	y4	0.1	0.9

Table 6.7: Transition function (T_y) of PayNet

Action	x_{t-1}	y_{t-1}	y_t			
			y1	**y2**	**y3**	**y4**
Auth	Any	y1	0.360	0.090	0.440	0.110
Auth	Any	y2	0.360	0.090	0.440	0.110
Auth	Any	y3	0.320	0.080	0.480	0.120
Auth	Any	y4	0.320	0.080	0.480	0.120
NoAuth	Any	y1	0.623	0.267	0.077	0.033
NoAuth	Any	y2	0.595	0.255	0.105	0.045
NoAuth	Any	y3	0.525	0.225	0.175	0.075
NoAuth	Any	y4	0.280	0.120	0.420	0.180

These transitions depend on the action (i.e., $Auth$, $NoAuth$) chosen in the current state.

Rewards

As previously discussed, each state in PayNet represents three factors: two of them signify the satisfaction level of NFRs, while the third one indicates the authenticity of the transacting customer. Thus, in PayNet, as shown in Table 6.8, each reward value is a three-dimensional vector r_{ijk} indicating the impact of an action on each factor. A higher reward value indicates higher favorability or priority. Conversely, a lower value signifies less favorability; if negative, it means the action is undesirable for that factor.

Table 6.8: Reward function of PayNet

State	Action	Reward vector values		
		r_{MINF}	r_{MAXS}	r_{CUST}
x1,y1	Auth	90	60	80
x1,y2	Auth	100	60	100
x1,y3	Auth	80	15	10
x1,y4	Auth	80	15	100
x2,y1	Auth	40	85	80
x2,y2	Auth	40	80	100
x2,y3	Auth	40	0	10
x2,y4	Auth	40	0	100
x1,y1	NoAuth	100	100	100
x1,y2	NoAuth	100	60	−100
x1,y3	NoAuth	100	100	100
x1,y4	NoAuth	80	80	−100
x2,y1	NoAuth	20	100	100
x2,y2	NoAuth	20	60	−100
x2,y3	NoAuth	20	100	100
x2,y4	NoAuth	20	60	−100

Observations

PayNet has five monitorables, namely SMAF24, SMAF168, SMAS24, SMAS168, and GEN. SMAF24 is a simple moving average (SMA) of the number of fraudulent transactions with a 24-hour window size, and SMAF168 is the 168-hour window size SMA. SMAS24 and SMAS168 are the SMAs for used-card transactions, respectively, with 24-hour window sizes and 168-hour window sizes. GEN is a boolean status of the authenticity of the customer returned by Algorithm 4 with 90% accuracy.

MINF is a fully observable state; hence, SMAF24 and SMAF168 directly dictate MINF's value. MINF is satisfied when SMA24 is lower than SMAF168, indicating that fraud can be minimised as the trend of fraud occurrences over 24 hours is lower than the average of the past seven days.

Meanwhile, MAXS is a partially observable state through the values of SMAS24 and SMAF168. If SMAS24 exceeds SMAF168, there is a positive trend in used-card transactions, which generally reflects the level of customer satisfaction with transactions via PayNet. However, this is only a probabilistic approach, as the actual level of customer satisfaction could be higher or lower.

Algorithm 4: The algorithm used for *GEN* to probe the authenticity of the customer

```
Input  : Transaction currency curr, Transaction amount amount
Output : The authenticity of the customer gen
if curr == 'EUR' and amount >=1 and amount <= 3200 then
    gen = False
else if curr == 'GBP' and amount >= 40 and amount <= 900 then
    gen = False
else if curr == 'USD' and amount >= 1 and amount <= 50 then
    gen = False
else
    gen = True
return gen
```

Table 6.9 presents conditional probabilities of MINF, MAXS, and CUST given different conditions of monitorables. Since PayNet utilises three monitorables with two boolean values each, it results in a total of eight observations with corresponding joint probabilities, as referenced in Table 6.9.

Table 6.9: Observation probabilities in PayNet

MON1	**SMA of the number of fraudulent transactions (SMAF24, SMAF168)**			
Action	P(MINF=T|SMAF24, SMAF168)		P(MINF=F|SMAF24, SMAF168)	
	SMAF24<=SMAF168	SMAF24>SMA168	SMAF24<=SMAF168	SMAF24>SMAF168
Auth	1	0	0	1
NoAuth	1	0	0	1
MON2	**SMA of used-card transactions (SMAS24, SMAS168)**			
Action	P(MAXS=T|SMAS24, SMAS168)		P(MAXS=F|SMAS24, SMAS168)	
	SMAS24<=SMAS168	SMAS24>SMA168	SMAS24<=SMAS168	SMAS24>SMAS168
Auth	0.35	0.65	0.65	0.35
NoAuth	0.20	0.80	0.80	0.20
MON3	**The genuineness of customers (GEN)**			
Action	P(CUST=T|GEN)		P(CUST=F|GEN)	
	GEN=T	GEN=F	GEN=T	GEN=F
Auth	0.9	0.1	0.1	0.9
NoAuth	0.9	0.1	0.1	0.9

6.3.6 Phase 6. Policy Planning

In this phase, we aim to obtain the best set of policies from the available model. Since the global weight vector is not determined in Phase 3, our current implementation of SPECTRA uses OLSAR and Perseus for offline planning. Once the process is complete, OLSAR presents us with a set of optimal weight vectors and their corresponding vector values and policies. With the optimal weight vectors available, we can select the best weight vector from the set. The selected optimal weight for PayNet is [0.1379, 0.7753, 0.0868].

6.4 Evaluation

This section discusses the experiment setup, baselines used for comparisons, and metrics measured to compare methods.

6.4.1 Experiment Setup

As seen in Figure 6.2, we employ similar digital twins architecture with the one we used for RDM in Chapter 5. Our evaluation uses the MultiMAuS simulator in the asset layer, which serves as the managed system. We connect it with the digital twin that acts as the managing system, running self-adaptation functions based on MR-MOMDP. However, despite both being based on the same architecture, the PayNet architecture differs from the RDM scenario in Chapter 5. The PayNet architecture includes humans in the asset layer (i.e., managed system), which is monitored by the digital twin (i.e., managing system).

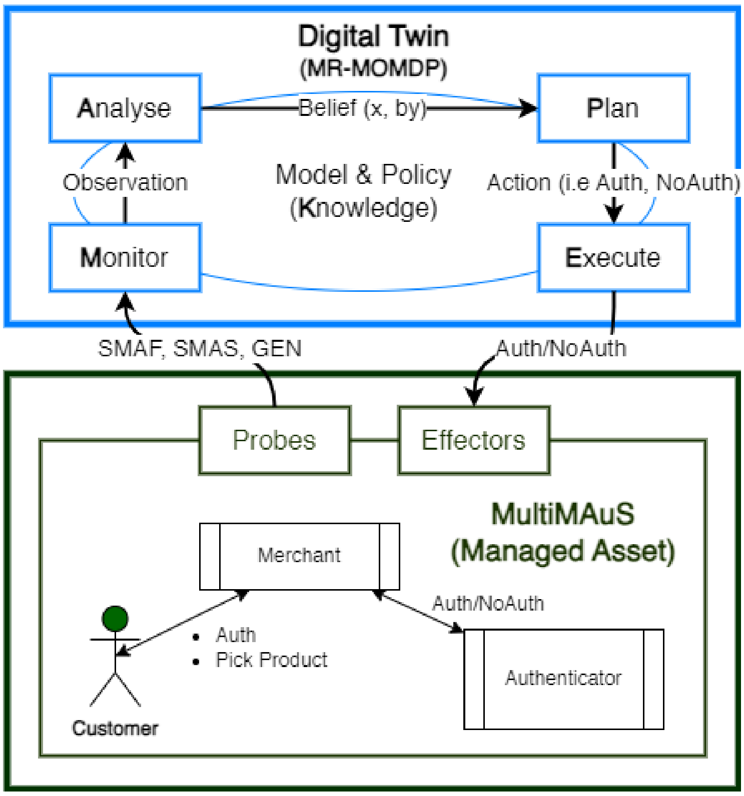


Figure 6.2: Digital twins architecture used in the evaluation with MultiMAuS and MR-MOMDP.

Table 6.10 contains all parameters used in the simulation, including various seed numbers, to provide variability and reproducibility. We used the period from June 1, 2016, to August 31, 2016, as summer is a peak time for transactions [32], and the intensity of fraud is also higher during this period.

Table 6.10: Configuration parameters for MultiMAuS

No	**Parameter**	**Value**
1	Random seed	1, 12, 123, 1234, 12345
2	Start date	01-Jun-16
3	End date	31-Aug-16
4	Noise level	0.1
5	Number of genuine customers	3333
6	Number of fraudsters	55
7	Initial satisfaction	1.0

6.4.2 Evaluation Metrics

Several metrics are being used in the evaluation to compare our approach vs the baselines, as follows:

- **Accuracy**: measuring the accuracy of authentication methods in identifying genuine and fraudulent customers involves deriving several sub-metrics such as *sensitivity*, *specificity*, etc.
- **Gross revenue** refers to the total revenue earned from successful genuine transactions.
- **Total loss** is the financial loss resulting from fraudulent transactions that require compensating the original credit card holder.
- **The net revenue** refers to the income left after subtracting losses from the gross revenue.
- **The action proportion** refers to the percentage of actions taken during the simulation period.
- **Constraint violation**: observing how each authentication mechanism may satisfy the given constraints.

We employ several sub-metrics to quantify accuracy, namely:

- False Positive: a genuine customer being incorrectly identified as a fraudster.
- False Negative: a fraudster is incorrectly identified as a genuine customer.
- True Positive: a fraudster is correctly identified as such.
- True Negative: a genuine customer is correctly identified as genuine.

- Sensitivity (True Positive Rate): the probability of correctly identifying a fraudster given that the individual is truly a fraudster; calculated using equation 6.1 [212].
- Specificity (True Negative Rate): the probability of correctly identifying a genuine customer given that the individual is truly genuine; calculated using equation 6.2 [212].

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \tag{6.1}$$

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}} \tag{6.2}$$

6.4.3 Baselines

We used the built-in authentication methods provided by MultiMAuS as baselines to evaluate our approaches, namely:

- *always_second*: the system always asks for a second authentication for every transaction [32].
- *oracle*: the ideal condition (i.e., utopia), the system knows precisely the genuity of the customers and hence only asks for a second authentication to the fraudsters [32].
- *heuristic*: The heuristic method follows a specific rule or criterion to decide

when to ask for a second authentication. Specifically, in MultiMAuS, the heuristic authenticator only asks for a second authentication if the transaction amount exceeds a certain threshold, such as 50 euros [32].

- *random*: The random authenticator asks for a second authentication randomly with a probability of 0.5 [32]. If the second authentication is provided, the transaction is authorised [32].

- *never_second*: the authenticator never asks for a second authentication, regardless of the transaction circumstances, it permits all incoming transactions without requesting further authentication [32].

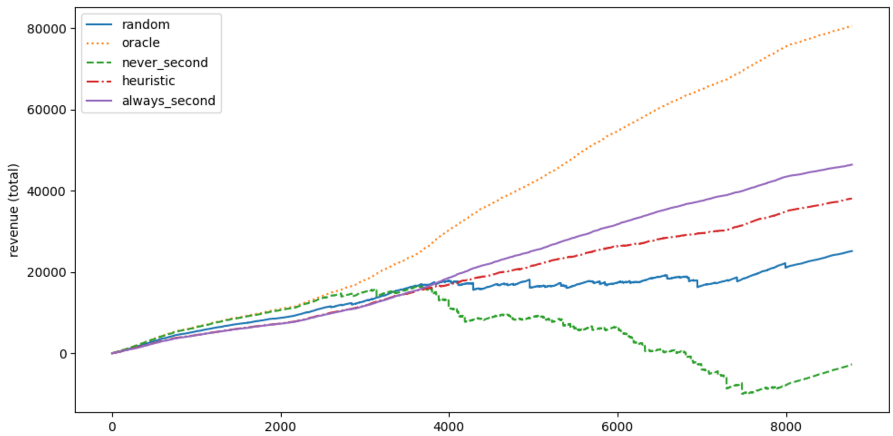


Figure 6.3: Total net revenue gained by each built-in authentication method of MuliMAuS

Figure 6.3 displays the various performances of MultiMAuS' built-in authentication methods. It is evident that the *never_second* method outperforms all the other methods, but it still has room for improvement. The *heuristics* method comes in second place. We identified an opportunity to enhance the existing heuristic method and named the improved version "dynamic". The "dynamic" approach, as in Algorithm 5, is a straightforward modification of an existing heuristic method, involving the refinement of the heuristic by incorporating additional rules. It does not involve user profiling, artificial intelligence, or the implementation of more fine-grained methods, which may be necessary. We modified *dynamic* as the observation algorithm for customer genuineness (GEN) in SPECTRA, as outlined in Algorithm 4.

Algorithm 5: The algorithm of *dynamic*

Input : Transaction currency *curr*, Transaction amount *amount*
Output : Boolean of authentication status *auth*

```
if curr == 'EUR' and amount >=1 and amount <= 3200 then
    auth = True
else if curr == 'GBP' and amount >= 40 and amount <= 900 then
    auth = True
else if curr == 'USD' and amount >= 1 and amount <= 50 then
    auth = True
else
    auth = random.choice([True, False], probability=[0.1, 0.9])
return auth
```

6.4.4 Results and Discussion

The following sections discuss the results of the experiments using the metrics we defined above.

Accuracy

Table 6.11 presents the accuracy of each authentication method under different random seed scenarios.

The *always_second* method exhibits the highest false positive rate, with false negatives and true negatives being zero, because it consistently requests a second authentication from the customer regardless of their authenticity. Therefore, *always_second* consistently scores 1 for sensitivity and 0 for specificity in all scenarios. The *heuristic* performs the worst, with the highest number of false negatives and the lowest number of true positives, resulting in the lowest sensitivity score. Conversely, because it is the most lenient compared to others, the *heuristic* has the highest number of true negatives, leading to the highest specificity as well.

Our modified *heuristic* (i.e., *dynamic*) refines the *heuristic* by adding more rules, significantly reducing false negatives. *Dynamic* exhibits good sensitivity in the range of 0.99-1, while its specificity is around 0.17. Implementing *dynamic* as part of our observation in SPECTRA can reduce false negatives and improve sensitivity, albeit with a tradeoff in its specificity. However, since PayNet aims to minimise fraud as much as possible, higher sensitivity is preferable.

Table 6.11: Accuracy of each authentication method under different random seeds

Random seed: 1						
Auth Method	False Positive	False Negative	True Positive	True Negative	Sensitivity	Specificity
always$_{s}$*econd*	26684	0	646	0	1.0000	0.0000
heuristic	17294	358	263	10962	0.4235	0.3880
dynamic	22743	1	664	4545	0.9985	0.1666
SPECTRA	22974	0	670	4172	1.0000	0.1537
Random seed: 12						
Auth Method	False Positive	False Negative	True Positive	True Negative	Sensitivity	Specificity
always$_{s}$*econd*	25790	0	603	0	1.0000	0.0000
heuristic	17055	356	234	10565	0.3966	0.3825
dynamic	21809	1	619	4736	0.9984	0.1784
SPECTRA	22169	0	625	4440	1.0000	0.1669
Random seed: 123						
Auth Method	False Positive	False Negative	True Positive	True Negative	Sensitivity	Specificity
always$_{s}$*econd*	26201	0	625	0	1.0000	0.0000
heuristic	6036	126	91	3678	0.4194	0.3786
dynamic	22264	0	607	4771	1.0000	0.1765
SPECTRA	22458	0	616	4480	1.0000	0.1663
Random seed: 1234						
Auth Method	False Positive	False Negative	True Positive	True Negative	Sensitivity	Specificity
always$_{s}$*econd*	26526	0	638	0	1.0000	0.0000
heuristic	17117	346	238	10742	0.4075	0.3856
dynamic	22552	1	653	4739	0.9985	0.1736
SPECTRA	22772	1	650	4423	0.9985	0.1626
Random seed: 12345						
Auth Method	False Positive	False Negative	True Positive	True Negative	Sensitivity	Specificity
always$_{s}$*econd*	26373	0	557	0	1.0000	0.0000
heuristic	17015	339	243	10878	0.4175	0.3900
dynamic	22509	4	530	4630	0.9925	0.1706
SPECTRA	22601	1	556	4302	0.9982	0.1599

Gross Revenue

Figure 6.4 illustrates the total gross revenue of different authentication methods under different random seed scenarios. As the most stringent, *always_second* yields the lowest total gross revenue. This is attributed to low customer satisfaction, resulting in fewer repeat transactions. With a more lenient policy, *heuristic* is capable of producing a fairly good total gross revenue, sometimes nearly matching *SPECTRA* or *dynamic*. *Dynamic* and *SPECTRA* produce total gross revenue with insignificant differences or even equivalently.

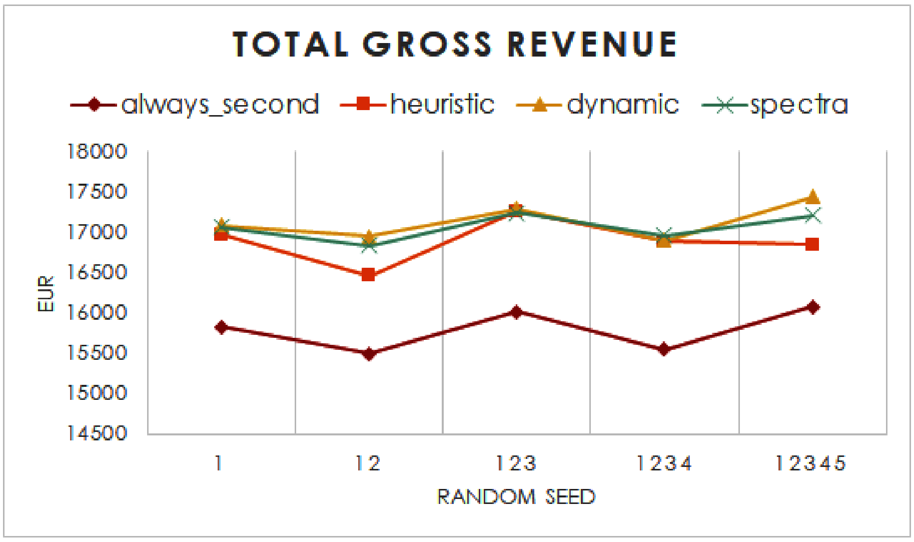


Figure 6.4: Total gross revenue on various authentication methods

Total Loss

The total loss is directly proportional to the false negatives generated by each method, indicating the number of successful fraudulent transactions. Figure 6.5 presents the total loss for various authentication methods across different random seed scenarios. Due to its constant request for second authentication in every transaction, *always_second* effectively eliminates total loss at the end of the period.

Meanwhile, *heuristic*, with its lenient policy, yields the worst total loss among the methods. *SPECTRA* outperforms *dynamic* in minimising loss due to a smaller number of successful fraudulent transactions. Even in scenario seed 12345, the total loss from dynamic surpasses *heuristic*.

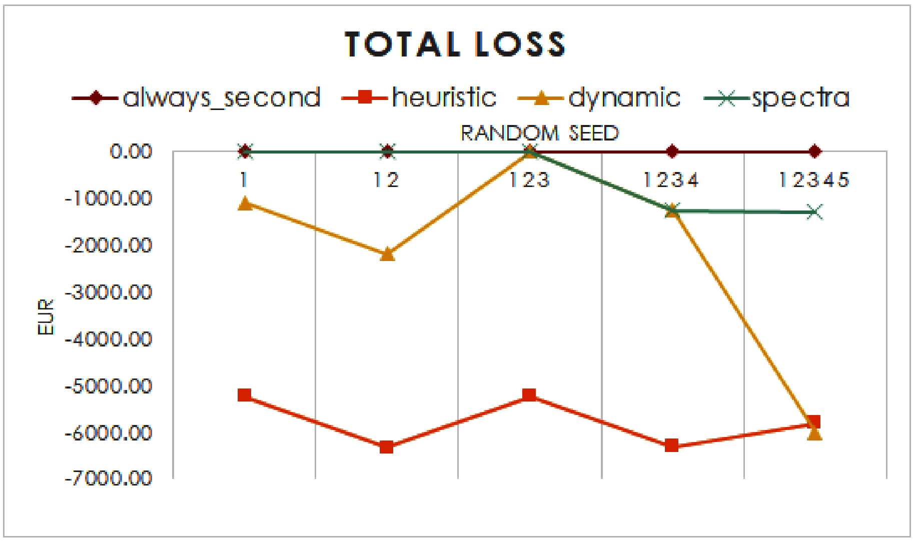


Figure 6.5: Total loss on various authentication methods

Net Revenue

Figure 6.6 illustrates the total loss for various authentication methods across different random seed scenarios. Our study found that *SPECTRA* consistently generates the highest total net revenue in most scenarios, while *heuristic* consistently performs the worst. *Dynamic* and *always_second* take turns outperforming each other in different scenarios. It is worth noting that *dynamic* performs the worst in seed 12345, with its total net revenue matching that of *heuristic*, which is the lowest level of performance in the comparison.

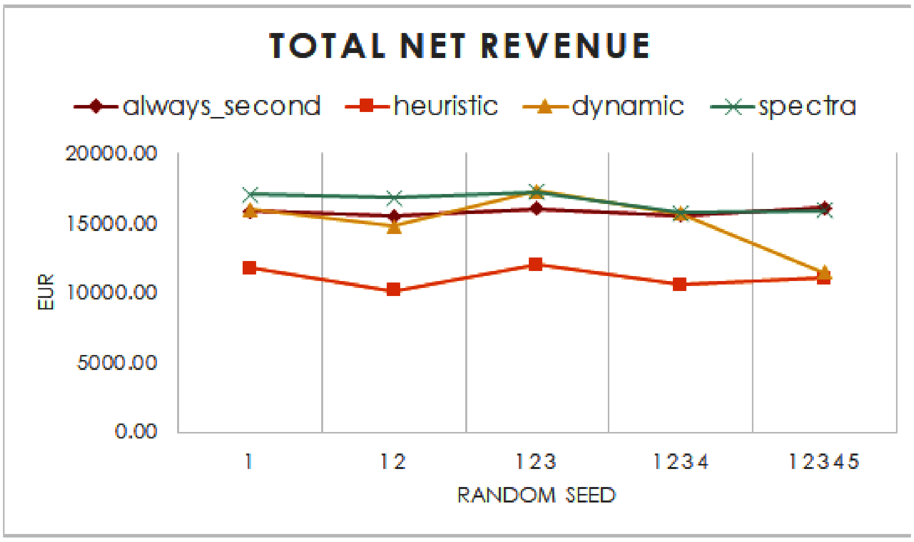


Figure 6.6: Total net revenue on various authentication methods

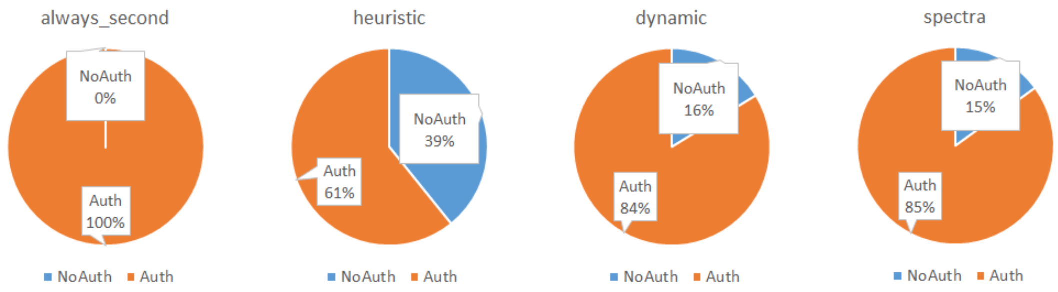
Action Proportion

Figure 6.7 illustrates the proportion of actions chosen in each transaction. As depicted in the figure, *always_second* consistently requests second authentications, resulting in 100% of actions chosen to be *Auth*. *Heuristic* opts for *NoAuth* in approximately 40% of transactions, yielding the highest total losses. *Dynamic* tightens its rules by selecting *Auth* for over 80% of transactions. *SPECTRA* allocates 1% more *Auth* than *dynamic*, leading to a decrease of approximately 1% in specificity compared to *dynamic*.

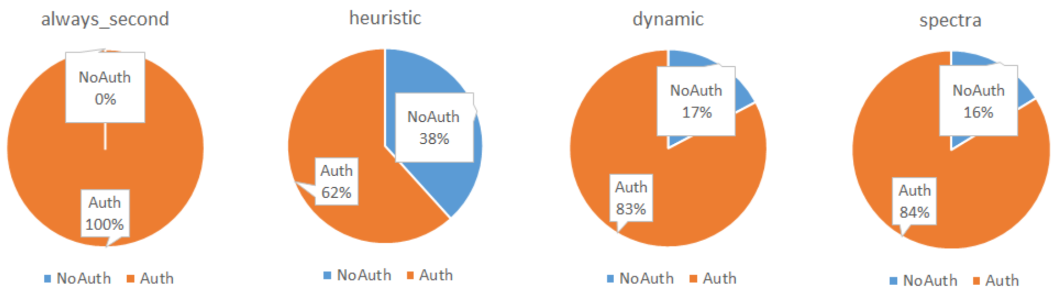
Constraint Violation

In this section, we assess constraint violation by introducing an ideal condition (i.e., utopia) labelled as *oracle* to gauge the extent to which existing approaches deviate from the ideal outcomes.

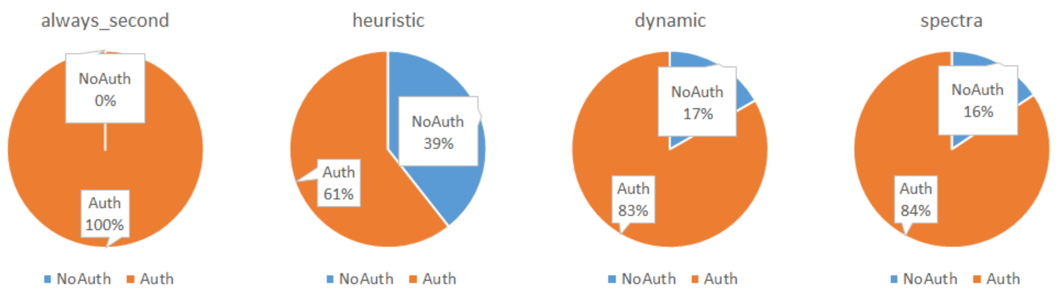
Figure 6.8 illustrates how each authentication method can satisfy their designated constraint for MAXS. As outlined in Table 6.4, we restrict the condition where



(a) Random seed: 1



(b) Random seed: 123



(c) Random seed: 12345

Figure 6.7: Action proportion in different authentication methods and different random seeds

SMAS24 < SMAS168 not to exceed 35%. PayNet employs a maximum threshold of 35% to provide flexibility, prioritising MINF in most instances.

It is apparent that in nearly all scenarios, only *always_second* surpasses the threshold. Conversely, *SPECTRA* and *dynamic* exceed the threshold only in scenario seed 12. This challenge could be addressed through improved techniques for identifying genuine customers, enabling *SPECTRA* and *dynamic* to reduce unnecessary authentications, thereby enhancing the number of repeat transactions.

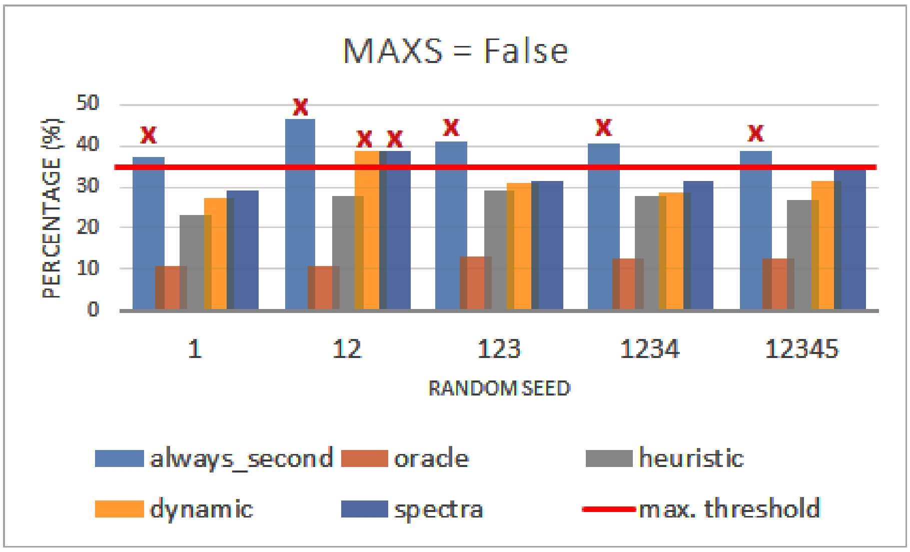


Figure 6.8: Constraint violation of Maximise Customer Satisfaction (MAXS)

Figure 6.9 presents the performance of each authentication method in maintaining the condition where SMAF24 > SMAS168 is less than %. All authentication methods, except *heuristic*, are capable of complying with the constraint. This is because *heuristic* is too lenient and fails to identify fraudsters effectively.

In most scenarios, the performance of *SPECTRA* can match *always_second* with zero MINF constraint violation, except in random seed 1234 and 12345. Nevertheless, it is evident that *SPECTRA* can better satisfy MINF compared to *dynamic*.

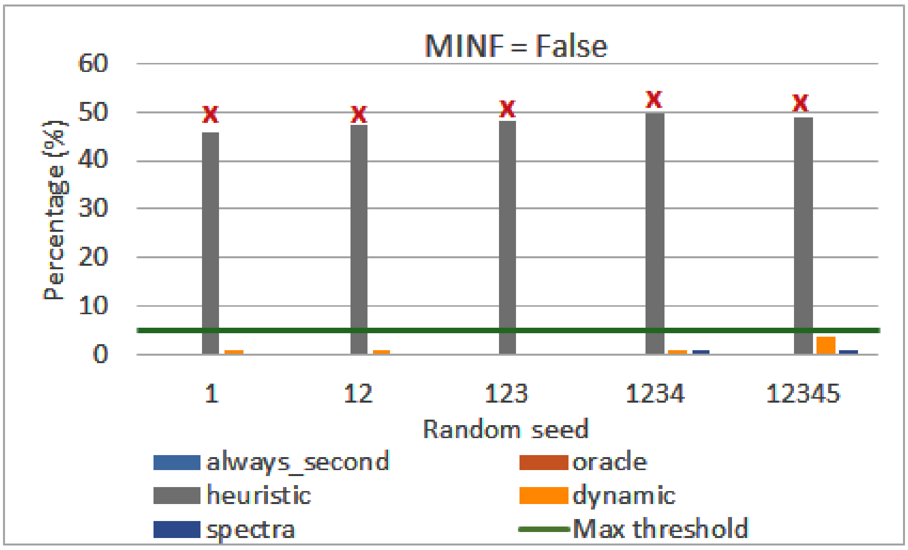


Figure 6.9: Constraint violation of Minimise Fraud (MINF)

6.5 Threats to Validity

6.5.1 Threats to External Validity

There are several factors that limit the generalisation of our research findings. First, our study was conducted within a digital twin environment with constrained settings and certain assumptions that may not fully represent real-world scenarios. Although MultiMAUS was built using a real-world dataset from a specific region and a limited time period, it may not fully reflect the behaviours of customers and fraudsters in other periods or regions. Thus, we collected data using five different random seed scenarios to introduce variability into our research.

6.5.2 Threats to Internal Validity

Inconsistencies in the simulation results can pose a threat to the internal validity of this research. We addressed this by using a fixed seed to ensure that any randomness introduced by the seed remains consistent across all iterations of the study.

Another potential threat is bias from random seed selection if the seed number is not chosen randomly or consistently or is chosen based on prior knowledge or preferences. To avoid this, we consistently selected five random seeds following the sequence pattern of 1, 12, 123, 1234, and 12345.

6.6 Conclusion

We have demonstrated the application of the SPECTRA framework within the context of HitLCPS in the credit card payment system domain facilitated by MultiMAuS, where humans are managed assets to be monitored to satisfy NFRs. We propose a refined version of the heuristic method and use it within the SPECTRA framework to provide a new authentication method that offers a better trade-off than existing methods provided by MultiMAuS. The evaluation reveals that the choice of authentication method significantly influences revenue generation, customer satisfaction, and the ability to prevent fraudulent activities.

SPECTRA has proven to provide a better trade-off than existing methods in most scenarios. SPECTRA exhibits high sensitivity while still providing opportunities to improve its specificity. Our current implementation only utilises rule-based techniques, but we believe that incorporating other AI techniques into SPECTRA for profiling genuine customers and transactions will yield even better results.

Chapter 7

Appraisal, Concluding Remarks, and Future Directions

This chapter presents a reflection and appraisal of the research, as well as several possible directions for future work. We begin by reviewing how the research questions listed in Chapter 1 have been addressed through our contributions. In the reflection, we discuss the feasibility of our proposed solutions, digital twins as an essential part of this research, and the computational overhead involved in the proposed solutions. At the end of this chapter, we discuss several future directions and wrap up with a conclusion.

7.1 Adressing Research Questions (RQ)

This section reviews how we have addressed our three research questions. We discuss our approach in tackling each question and present the findings that contribute to our overall understanding of the research topic.

7.1.1 Research Question 1

RQ1. *Realising that humans are generally placed as operators (i.e., service consumers) within Cyber-Physical Systems (CPS)* - **How can the human-machine collaboration in CPS be engineered for self-adaptation** that involves support from humans and machines? What human aspects/properties should be considered?

In Chapter 3, we introduce a classification of human-as-a-service within CPS and propose an SOA ontology model for the CPS environment as part of the Everything-as-a-Service paradigm. This model utilises the OWL-S and O*NET frameworks as primary references for building a human capability model that puts humans not only as service consumers but also as service providers.

We identify three key aspects of effective human-machine service provisioning: task, context, and capability. This means that to allocate tasks optimally, we must have resources (i.e., machines and humans) with appropriate context and capability. Our ontology defines human capability through three classes: characteristics, qualifications, and potential. We categorise human characteristics into preferences, abilities, and performance factors, encompassing work values (e.g., fairness, ethics) and style.

We detail how the MAPE-K model can be refined to leverage the advantages of this approach. Through use cases in the medical domain, we demonstrate the feasibility of our conceptual reference model and highlight how effective human-machine service provision can be achieved.

Our evaluation suggests that our ontology provides a high level of accuracy, consistency, and adaptability while achieving greater clarity and completeness than the baseline approaches. The proposed ontology supports both human-machine service provisioning and individual provisioning by either humans or machines. Furthermore, existing concepts in our ontology can be leveraged for additional applications beyond service provisioning. Chapter 3 draws from the work presented in [27].

7.1.2 Research Question 2

RQ2. *Realising that humans are value-driven and fairness is essential for fostering human-machine collaboration* - **How can human-machine service provisioning be continuously adapted to remain optimal considering human values (sc., fairness) and the constraints of humans and machines?**

Among the various human values, fairness stands out as one of the most significant. In traditional workplaces, fairness is widely acknowledged as a key driver of human motivation, loyalty, and productive collaboration. Therefore, in Chapter 4, we present our contribution: a fairness-aware task allocation approach for heterogeneous workers leveraging digital twins.

We use the case of crowdsourced delivery involving both human and machine workers, consisting of several groups of human workers and one group of autonomous vehicles (AVs), each with different capabilities and spatial constraints. Additionally, each task has a temporal constraint that must be met for successful completion.

We adopt social psychology's three principles of distributive justice: equity, equality, and need. We develop an unfairness index for each principle and integrate it into our task allocation algorithm, which utilises maximum-weight bipartite matching. This approach allows us to optimally allocate tasks to suitable workers who meet the spatio-temporal constraints while ensuring fairness. We use an evolutionary approach on the digital twin to adjust the weights for equity, equality, and need, enabling runtime adaptation.

We also use the digital twin architecture to analyse system behaviour and conduct evaluations using various incentive scenarios commonly found in real-world cases. Our approach has shown improved overall fairness outcomes in multiple experiments across different scenarios compared to the baseline methods. The evolutionary approach allows continuous adaptation that keeps human-machine service provisioning optimal. Chapter 4 draws partially from the work presented in [28].

7.1.3 Research Question 3

RQ3. Given that uncertainties can potentially breach the NFRs of HitLCPS and necessitate a trade-off, how can these uncertainties be anticipated and the trade-offs dynamically optimised during service provisioning?

One prominent type of uncertainty that presents a significant obstacle during runtime decision-making is epistemic uncertainty, which arises from insufficient knowledge. In the context of NFR satisfaction, it is often the case that not all NFR metrics can be fully observed; some may only be partially observed.

In Chapter 5, we propose SPECTRA, a framework designed to handle trade-offs among NFRs in dynamic environments with varying levels of NFR observability. This framework is based on a Markov decision process (MDP) that uses a vector reward to represent the priority or favourability of each objective (e.g., NFR satisfaction) in each state.

Additionally, we present MR-MOMDP, which extends beyond traditional MDP approaches to incorporate multiple objectives in mixed-observability settings.

We evaluate our framework using a digital twin architecture in two different use cases. In Chapter 5, we use Remote Data Mirroring scenarios, and in Chapter 6, we explore a credit card payment system scenario. The evaluation results indicate that our approach can achieve higher utility values, reduce the time required for policy planning, offer better trade-offs, and more effectively satisfy the NFRs. Chapter 5 was derived in part from the work presented in [31]

7.2 Reflections on the Research

This section presents our reflections on the research and the contributions made, evaluated through the feasibility of the proposed solutions, digital twins, and computational overhead.

7.2.1 Feasibility of the Proposed Solutions

As previously outlined, our proposed SOA-HitLCPS ontology, discussed in Chapter 3, serves as a reference model for the development of HitLCPS, where humans can act as service consumers and/or service providers. We have demonstrated its feasibility using a use case from the medical domain. Our analysis indicates that our ontology can be further extended or may need to be adapted to suit the specific requirements of different domains.

Our fairness-aware task allocation approach in Chapter 4 emphasises the importance of considering equity, equality, and need in human-machine service provisioning. We utilise several incentive scenarios drawn from real-world cases to ensure our approach is grounded in practical applications. Consequently, our unfairness indexes and algorithms can be applied to comparable crowdsourced delivery scenarios. However, for different cases, the unfairness indexes and algorithms we propose may require modification to suit the specific domain.

SPECTRA is a framework for specifying NFRs and building MDP models in accordance with the observability of each NFR (and other supporting variables). This framework provides flexibility so that one can develop MDP models tailored to the work environment. We have demonstrated the use of SPECTRA in two distinct domains -remote data mirroring (using RDMSim) and credit card payment systems (using MultiMAuS)— to showcase this flexibility.

Many HitLCPS scenarios involve humans working with UAVs, robots, and other AI agents in various fields such as manufacturing, transportation, and military operations. However, publicly available datasets are very limited, if any. There are also no open-source simulators available other than RDMSim and MultiMAuS. However, using these two simulators may pose a threat to construct validity, for these simulators can be built with oversimplification that does not capture confounding variables that may exist in real-world applications and other scenarios. Evaluating SPECTRA with only these two simulators also poses threats to external validity, limiting the generalisability of our findings to different settings. However, we minimise it using various scenarios and several random seed numbers.

While the MirrorNet (i.e., RDMSim) and PayNet (i.e., MultiMAuS) scenarios may not explicitly involve human operators, assuming the presence of a human supervisor in MirrorNet operation allows us to categorise this system as HitLCPS (i.e., human-on-the-loop; see Chapter 2). Furthermore, our credit card payment system (i.e., PayNet) scenario is a HitLCPS due to its explicit involvement of humans as service customers; this system monitors their transactions and optimises their satisfaction.

Although our evaluations may have threats to external validity, construct validity, and internal validity (as we have explained in Chapter 5 and Chapter 6), the use of different scenarios to evaluate SPECTRA has convinced us of the potential it holds. We are confident that SPECTRA can assist other self-adaptive systems across various domains and with different types of observability, opening up a world of future possibilities.

This thesis contributes to multiple areas, demonstrating how our proposed solutions can effectively apply to different use cases. Our approaches showcase the potential for adaptation and flexibility across various domains of human-machine collaboration.

7.2.2 Digital Twins

Digital twin architecture is an essential aspect of this research. We use two levels of digital twins, among the levels described in Section 2.3.6, to instantiate and evaluate our approach.

All digital twins deployed in this research are implemented on a laptop equipped with an Intel(R) Core(TM) i3-5005U CPU running at 2.00 GHz and 8 GB of RAM. The laptop runs the Windows 10 Pro operating system.

In Chapter 4, we propose an architecture for crowdsourced delivery and instantiate it at both design time and runtime to enable self-adaptation. Our digital twins are developed in a Python environment using SimPy, a process-based discrete-event simulation framework. During the design phase, we instantiate our digital twins' architecture as a Digital Blueprint to explore the design space and understand system behaviour across various incentive scenarios. At runtime, we instantiate the reference architecture as a Pure-Play to oversee the assets and accomplish system objectives through adaptation. In the absence of real-world datasets, we use synthetic datasets in our experiments. This approach allows us greater flexibility in conducting what-if analyses and exploring scenarios that may not be captured in real-world datasets. Nonetheless, we acknowledge that access to real-world datasets would aid in validating our proposals.

Chapters 5 and 6 use a similar digital twin architecture, where we position existing simulation software as the managed asset and apply SPECTRA to implement Pure-Play, a dominant digital twin, as the managing system.

In Chapter 5, we use RDMSim, a simulator that enables researchers to evaluate and compare decision-making techniques for self-adaptation in the context of Remote Data Mirroring, providing realistic scenarios and data for benchmarking under environmental uncertainty.

In Chapter 6, our work is based on MultiMAuS, an agent-based simulator designed for payment transactions, developed specifically for the analysis and development of dynamic online fraud detection methods utilising a multi-modal user authentication system. It uses real-world credit card transaction data to accurately model customer behaviour.

SPECTRA, implemented at the managing system operating at the digital twin layer, generates policies and makes runtime decisions using the policies to optimise long-term rewards.

Despite conducting these evaluations in settings that were made as realistic as possible, digital twins remain in a controlled environment that may obscure certain environmental dynamics present in real-world implementations. Consequently, we acknowledge the need for further research to assess the effectiveness of our approach in a real-world HitLCPS environment.

7.2.3 Computational Overhead

In this section, we address computational overhead in two parts. First, we discuss the computational overhead associated with bipartite graphs and the evolutionary approach in our fairness-aware task allocation mechanism presented in Chapter 4. Following that, we review the computational overhead in our SPECTRA framework, which was introduced in Chapters 5 and 6.

Impact of graph size and evolutionary approach on our fairness-aware task allocation

In the non-adaptive solution we propose in Chapter 4 (i.e., FMWM), the source of computational overhead stems from the number of tasks and workers, which determines the number of vertices in the bipartite graph. This computational overhead should not pose a significant challenge at a smaller scale and with an appropriate algorithm. Our dynamic evolutionary approach (i.e., Adaptive FMWM) incurs computational overhead from the population size and the number of generations used. In our scenarios, up to five generations and a population size of five was sufficient to find a near-optimal solution. However, different settings may require larger populations and more generations, resulting in a trade-off with longer processing times.

Impact of observability on SPECTRA

SPECTRA, as discussed in Chapters 5 and 6, presents varying computational overhead depending on the number of states, actions, and the observability of possible states. As previously mentioned, SPECTRA is a framework based on multi-reward Markov models (i.e., MDP, POMDP, MOMDP). The computational overhead of solving MDP, POMDP, and MOMDP varies significantly due to the complexity of the problems they represent.

Solving POMDPs generally incurs the highest computational overhead due to the continuous belief state space (i.e., the probability distribution over possible states) that must be maintained and the partial observability of the environment. The complexity of POMDPs can be exponential in the state and action spaces, making them computationally intractable for large problems.

MOMDPs offer an intermediate level of complexity, taking advantage of the factorization of state space into fully observable and partially observable states to enhance efficiency. The computational complexity of MDP depends on the size of the state and action spaces; however, with its fully observable state, MDP usually has the lowest overhead and is, therefore, the most tractable among the three.

Using vector rewards in SPECTRA (i.e., MR-MDP, MR-MOMDP, and MR-POMDP) adds an additional layer of complexity due to the need to account for multiple objectives. First, the use of vector rewards necessitates a more complex policy representation (i.e., converting AlphaVector in POMDP to AlphaMatrix in MR-POMDP). Second, multi-objective optimisation is an additional process that often requires numerous iterations to discover Pareto optimal policies, thus taking much longer to solve problems with vector rewards compared to scalar rewards.

Therefore, applying SPECTRA to problems with large partially observable states and a high number of objectives can be computationally prohibitive. We recommend using SPECTRA for problems with limited partially observable states and a constrained number of objectives.

7.3 Implications and Future Directions

The findings from this thesis offer foundational insights that shape future research in human-machine collaboration by defining core areas where additional investigation and development are essential. These findings can drive future research agendas in the following areas:

7.3.1 Extension of SOA-HitLCPS ontology

The proposed SOA-HitLCPS ontology serves as a reference model and adaptable framework that can be customised for various application areas. This flexibility implies that researchers and practitioners can use the ontology as a blueprint for developing human-machine systems that are optimised for specific domains, including but not limited to healthcare, military, manufacturing, logistics, and transportation.

Consistent with our findings in Chapter 3, our SOA-HitLCPS ontology provides adaptability, enabling extension, integration, and adaptation. This ontology can be further developed and tailored for use across various domains involving both human and machine workers, and it also supports scenarios involving either human-only or machine-only services without the necessity for both to be present. Although originally designed for SOA, our ontology serves as a general reference model for

HitLCPS. As we have demonstrated in Chapters 4, 5, and 6, one can adopt SOA entirely or partially, incorporating it into non-SOA models (e.g., holon [137])as needed. Ontology-based approach is enabling autonomous discovery and composition of the systems; it simplifies the composition and adaptation of heterogeneous systems in dynamic environments [138]. In Chapter 3, we outline that concepts such as $Machine\_Specification$ and $Experience$ need to be refined further to anticipate failures both predictively and proactively. Future research can focus on refining this ontology for sector-specific needs and expanding it to accommodate emerging fields and address evolving ethical and legal requirements in different contexts of human-robot collaboration.

7.3.2 Considering Fairness and other HitLCPS Dynamics

Addressing fairness remains a key priority for building trust in human-machine systems. This thesis' fairness-aware task allocation models offer a starting point for developing algorithms that incorporate fairness principles that can help in task allocation and resource sharing for a heterogeneous setting of human-machine collaboration. These considerations have broad implications, from maintaining workforce morale in automated environments to enhancing public trust in autonomous systems in public services

In line with our findings in Chapter 4, further research on fairness, particularly in the area of human-machine collaboration, is necessary. While we have introduced equity, equality, and need as metrics in our computations, there are still many opportunities to explore these concepts in other domains and use cases. In this study, we only worked with a single variable to define need; however, the need could involve multiple dynamic variables (e.g., fuel consumption, fatigue, traffic conditions, etc) that may change over time and may require shifting priorities.

Moreover, in Chapter 4, we employed multi-objective approaches focusing solely on fairness. In many other real-world settings, it may be necessary to incorporate other dynamics of HitLCPS and additional factors into the objectives. For example, in use cases that optimise revenue and service reliability, but also ensure worker safety while maintaining fairness. Various decision-making techniques, including the SPECTRA framework presented in Chapter 5, and also AI, could serve as directions for future exploration.

7.3.3 Improving Policy Planning of SPECTRA

The SPECTRA framework for integrating NFRs into MDP models offers a meaningful step forward in refining decision-making processes for self-adaptive systems. SPECTRA highlights different types of observability of NFRs in SAS and HitLCPS and presents appropriate MDP models to solve the problem. Our findings suggest accuracy of the model as well as weight vectors and reward vectors greatly determines the quality of the resulting policy.

In line with our findings in Chapter 5 and 6, the values of weight vectors and reward vectors significantly impact system behaviour. While we have solutions within the Pareto set, it is not guaranteed that every solution will yield behaviour that aligns with our expectations. The current implementation of SPECTRA still relies on human judgement to select an optimal weight vector from the Pareto set that yields the desired behaviour. This aspect presents an opportunity for further refinement to enhance the efficiency and effectiveness of this stage.

Moreover, our current implementation utilises the Perseus solver for offline planning tasks. While Perseus serves its purpose well, alternative solvers like SARSOP [128, 213] may offer better performance in specific scenarios. Additionally, considering the dynamic nature of many real-world environments, exploring the application of online planning techniques using solvers such as DESPOT [214] could be highly beneficial, especially for large problems. This would allow for more adaptive and responsive decision-making in dynamic settings.

Market dynamics, technological advancements, and regulatory changes may require adjustments and evolution of NFRs. The current version of SPECTRA does not anticipate the evolution of NFRs at runtime. Changes to NFRs necessitate redefining the state and creating a new model, requiring the system to be retrained to generate updated policies. If changes to NFRs are expectable, they can be anticipated by factoring state representations to accommodate potential future NFRs. However, if these adjustments lead to an excessively large state space, the problem may become intractable. If the dynamics only affect changes to the transition function and reward function, then exploring non-stationary MDPs [215] and online learning [216] could be valuable areas for future research.

7.4 Concluding Remarks

We have presented multiple solutions for self-adaptation in HitLCPS, instantiated, and evaluated them using various types of Digital Twins.

An ontology has been proposed as a reference model for service composition and task allocation in HitLCPS. Our ontology was demonstrated and evaluated using medical scenarios and assessed based on clarity, adaptability, consistency, completeness, and accuracy. Results show that the ontology is feasible and applicable in various domains and use cases.

Realising the motivational aspects of human behaviour and the significance of fairness, we advocate for equity, equality, and need in ensuring fair task allocation for both humans and machines. We have illustrated how these principles can be quantified and showcased their implementation alongside spatiotemporal constraints through bipartite graph matching in task allocation scenarios involving humans and machines. Moreover, our research demonstrates that using an evolutionary approach to runtime adaptation can enhance overall fairness. We hope our work motivates further research that applies equity, equality, and need in HitLCPS.

We have developed a framework that helps specify NFRs by using Markovian approaches to model problems based on the observability level of the environment. Our findings suggest that using a Markov model that aligns with the observability level of the environment can accelerate policy planning, provide higher expected total rewards, and offer better trade-offs. However, we need to futher explore to improve the weight selection process and provide online planning for more dynamic environments.

It is worth noting that Digital Twins enable us to perform analysis and planning at design time and provide self-adaptation at runtime, which could better satisfy the NFRs. However, we acknowledge that there are still many environmental dynamics that we have not accounted for in this study. Therefore, we greatly appreciate further research investigating our proposed approaches in other contexts, particularly in real-world settings.

Bibliography

[1] D. Weyns, "Software engineering of self-adaptive systems," *Handbook of software engineering*, pp. 399–443, 2019.

[2] C. Krupitzer, F. M. Roth, M. Pfannemüller, and C. Becker, "Comparison of approaches for self-improvement in self-adaptive systems," in *2016 IEEE International Conference on Autonomic Computing (ICAC)*, 2016, pp. 308–314.

[3] M. M. Cummings, "Man versus machine or man+ machine?" *IEEE Intelligent Systems*, vol. 29, no. 5, pp. 62–69, 2014.

[4] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. 46, no. 1, pp. 36–45, 2013.

[5] J.-B. Mouret and K. Chatzilygeroudis, "20 years of reality gap: a few thoughts about simulators in evolutionary robotics," in *Proceedings of the genetic and evolutionary computation conference companion*, 2017, pp. 1121–1124.

[6] "O∗NET OnLine," Nov 2019, [accessed 10. Nov. 2020]. [Online]. Available: https://www.onetonline.org

[7] P. M. Fitts, "Human engineering for an effective air-navigation and traffic-control system." 1951.

[8] H. Muccini, M. Sharaf, and D. Weyns, "Self-adaptation for cyber-physical systems: a systematic literature review," in *Proceedings of the 11th international*

symposium on software engineering for adaptive and self-managing systems, 2016, pp. 75–81.

[9] M. Afanasov, L. Mottola, and C. Ghezzi, "Towards context-oriented self-adaptation in resource-constrained cyberphysical systems," in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, 2014, pp. 372–377.

[10] S. K. Sowe, E. Simmon, K. Zettsu, F. de Vaulx, and I. Bojanova, "Cyber-Physical-Human Systems: Putting People in the Loop," *IT Professional*, vol. 18, no. 1, pp. 10–13, Jan 2016.

[11] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an internet of all," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 944–965, 2015.

[12] W. Damm, D. Hess, M. Schweda, J. Sztipanovits, K. Bengler, B. Biebl, M. Fränzle, W. Hagemann, M. Held, K. Ihme *et al.*, "A reference architecture of human cyber-physical systems–part i: Fundamental concepts," *ACM Transactions on Cyber-Physical Systems*, vol. 8, no. 1, pp. 1–32, 2024.

[13] A. Bennaceur, D. Hassett, B. Nuseibeh, and A. Zisman, "Values@ runtime: An adaptive framework for operationalising values," in *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society*. IEEE, 2023, pp. 175–179.

[14] M. A. Ferrario and E. Winter, "Applying human values theory to software engineering practice: Lessons and implications," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 973–990, 2023.

[15] A. Sutcliffe, P. Sawyer, and N. Bencomo, "The implications of 'soft'requirements," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*. IEEE, 2022, pp. 178–188.

[16] J. Whittle, M. A. Ferrario, W. Simm, and W. Hussain, “A case for human values in software engineering,” *IEEE Software*, vol. 38, no. 1, pp. 106–113, 2019.

[17] E. Winter, S. Forshaw, and M. A. Ferrario, “Measuring human values in software engineering,” in *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, 2018, pp. 1–4.

[18] H. Perera, W. Hussain, J. Whittle, A. Nurwidyantoro, D. Mougouei, R. A. Shams, and G. Oliver, “A study on the prevalence of human values in software engineering publications, 2015 – 2018,” in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, p. 409–420.

[19] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, “Industry 4.0 and Industry 5.0—Inception, conception and perception,” *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, Oct. 2021.

[20] E. Valette, H. Bril El-Haouzi, and G. Demesure, “Industry 5.0 and its technologies: A systematic literature review upon the human place into IoT- and CPS-based industrial systems,” *Computers & Industrial Engineering*, vol. 184, p. 109426, Oct. 2023.

[21] D. Hüsener, M. Schluse, D. Kaufmann, and J. Roßmann, “The digital twin as a mediator for the digitalization and conservation of expert knowledge,” *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*, p. 241, 2022.

[22] P. Kalina, “Why do we work? motive portends outcome,” *International Journal of Business Marketing and Management*, vol. 6, no. 4, pp. 39–41, Apr 2021.

[23] R. M. Ryan and E. L. Deci, “Self-determination theory and the facilitation of

intrinsic motivation, social development, and well-being." *American Psychologist*, vol. 55, pp. 68–78, 2000.

[24] X. Tao, D. G. Broo, M. Törngren, and D. Chen, "Uncertainty management in situation awareness for cyber-physical systems: State of the art and challenge," in *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, 2020, pp. 424–430.

[25] "OWL-S: Semantic Markup for Web Services," Nov. 2004, [accessed 23. Aug. 2020. [Online]. Available: https://www.w3.org/Submission/OWL-S

[26] S. C. f. B. I. Research, "protégé," Dec 2020, [accessed 12. Dec. 2020]. [Online]. Available: https://protege.stanford.edu/products.php#desktop-protege

[27] H. T. N. Ignatius and R. Bahsoon, "A Conceptual Reference Model for Human as a Service Provider in Cyber Physical Systems," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE, May 2021, pp. 1–10.

[28] ——, "Equity, equality, and need: Digital twin approach for fairness-aware task assignment of heterogeneous crowdsourced logistics," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 3, pp. 3420–3431, 2024.

[29] H. Samin, N. Bencomo, and P. Sawyer, "Decision-making under uncertainty: be aware of your priorities," *Software and Systems Modeling*, vol. 21, no. 6, pp. 2213–2242, 2022.

[30] H. Soh and Y. Demiris, "Evolving policies for multi-reward partially observable markov decision processes (mr-pomdps)," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 713–720.

[31] H. T. N. Ignatius, H. Samin, R. Bahsoon, and N. Bencomo, "Spectra: a markovian framework for managing non-functional requirements tradeoffs with di-

verse levels of observability in self-adaptive software systems," *ACM Transactions on Autonomous and Adaptive Systems*, Submitted: 30 September 2023.

[32] L. M. Zintgraf, E. A. Lopez-Rojas, D. M. Roijers, and A. Nowé, "Multimaus: a multi-modal authentication simulator for fraud detection research," in *29th European Modeling and Simulation Symposium (EMSS)*, 2017, pp. 360–370.

[33] R. Bahsoon, "Acm transactions on autonomous and adaptive systems (acm taas): Editorial welcome and update on state of the journal, vision, and ongoing developments," *ACM Transactions on Autonomous and Adaptive Systems*, 2024.

[34] O. Gheibi and D. Weyns, "Lifelong self-adaptation: Self-adaptation meets lifelong machine learning," in *Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2022, pp. 1–12.

[35] "Cyber-Physical Systems (CPS) (nsf20563) | NSF - National Science Foundation," Apr 2020, [accessed 10. Apr. 2020]. [Online]. Available: https://www.nsf.gov/pubs/2020/nsf20563/nsf20563.htm

[36] "Cyber-Physical Systems - a Concept Map," Apr 2020, [accessed 10. Apr. 2020]. [Online]. Available: https://ptolemy.berkeley.edu/projects/cps

[37] B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Di Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. M. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, and J. Whittle, "Software Engineering for Self-Adaptive Systems: A Research Roadmap," in *Software Engineering for Self-Adaptive Systems*. Berlin, Germany: Springer, 2009, pp. 1–26.

[38] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research

challenges," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, pp. 1–42, 2009.

[39] A. Elhabbash, M. Salama, R. Bahsoon, and P. Tino, "Self-awareness in software engineering: A systematic literature review," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 14, no. 2, pp. 1–42, 2019.

[40] T. Chen, R. Bahsoon, and X. Yao, "A survey and taxonomy of self-aware and self-adaptive cloud autoscaling systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–40, 2018.

[41] F. Faniyi, P. R. Lewis, R. Bahsoon, and X. Yao, "Architecting self-aware software systems," in *2014 IEEE/IFIP Conference on Software Architecture.* IEEE, 2014, pp. 91–94.

[42] M. Blumreiter, J. Greenyer, F. J. Chiyah Garcia, V. Klös, M. Schwammberger, C. Sommer, A. Vogelsang, and A. Wortmann, "Towards self-explainable cyber-physical systems," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 543–548.

[43] A. Computing *et al.*, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, no. 2006, pp. 1–6, 2006.

[44] R. De Lemos, H. Giese, H. A. Muller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II.* Springer, 2013, pp. 1–32.

[45] S. Zeadally, T. Sanislav, and G. D. Mois, "Self-adaptation techniques in cyber-physical systems (cpss)," *IEEE Access*, vol. 7, pp. 171 126–171 139, 2019.

[46] A. Musil, J. Musil, D. Weyns, T. Bures, H. Muccini, and M. Sharaf, “Patterns for self-adaptation in cyber-physical systems,” in *Multi-disciplinary engineering for cyber-physical production systems.* Springer, 2017, pp. 331–368.

[47] F. M. Zanzotto, “Human-in-the-loop artificial intelligence,” *Journal of Artificial Intelligence Research*, vol. 64, pp. 243–252, 2019.

[48] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, “Cyber-physical-social systems: A state-of-the-art survey, challenges and opportunities,” *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 389–425, 2020.

[49] J. Rasmussen, “Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, pp. 257–266, 1983.

[50] A. Lavazza, “Human Beings and Robots: A Matter of Teleology?” *HUMANA.MENTE Journal of Philosophical Studies*, vol. 11, no. 34, pp. 17–39, Dec 2018.

[51] N. Li, S. Adepu, E. Kang, and D. Garlan, “Explanations for human-on-the-loop: A probabilistic model checking approach,” in *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2020, pp. 181–187.

[52] J. Cleland-Huang, A. Agrawal, M. Vierhauser, M. Murphy, and M. Prieto, “Extending mape-k to support human-machine teaming,” in *Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2022, pp. 120–131.

[53] L. Wang, Z. Yu, K. Wu, D. Yang, E. Wang, T. Wang, Y. Mei, and B. Guo, “Towards robust task assignment in mobile crowdsensing systems,” *IEEE Transactions on Mobile Computing*, 2022.

[54] Z. Wang, C. Guo, J. Liu, J. Zhang, Y. Wang, J. Luo, and X. Yang, "Accurate and privacy-preserving task allocation for edge computing assisted mobile crowdsensing," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 120–133, 2022.

[55] L. Metcalf, D. A. Askay, and L. B. Rosenberg, "Keeping Humans in the Loop: Pooling Knowledge through Artificial Swarm Intelligence to Improve Business Decision Making," *California Management Review*, vol. 61, no. 4, pp. 84–109, Jul. 2019.

[56] D. C. McClelland, "How motives, skills, and values determine what people do." *American psychologist*, vol. 40, no. 7, p. 812, 1985.

[57] S. Spiekermann and T. Winkler, "Value-based engineering with ieee 7000," *IEEE Technology and Society Magazine*, vol. 41, no. 3, pp. 71–80, 2022.

[58] N. Zhang, R. Bahsoon, and G. Theodoropoulos, "Towards Engineering Cognitive Digital Twins with Self-Awareness," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 11–14.

[59] H. Feng, C. Gomes, S. Gil, P. H. Mikkelsen, D. Tola, P. G. Larsen, and M. Sandberg, "Integration of the mape-k loop in digital twins," in *2022 Annual Modeling and Simulation Conference (ANNSIM)*, 2022, pp. 102–113.

[60] A. Bilberg and A. A. Malik, "Digital twin driven human–robot collaborative assembly," *CIRP Annals*, vol. 68, no. 1, pp. 499–502, Jan. 2019.

[61] S. Yun, J.-H. Park, and W.-T. Kim, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017, pp. 922–926.

[62] A. Agrawal, R. Thiel, P. Jain, V. Singh, and M. Fischer, “Digital twin: Where do humans fit in?” *Automation in Construction*, vol. 148, p. 104749, 2023.

[63] L. F. Rivera, M. Jiménez, N. M. Villegas, G. Tamura, and H. A. Müller, “Toward autonomic, software-intensive digital twin systems,” *IEEE Software*, vol. 39, no. 2, pp. 20–26, 2022.

[64] J. Parri, F. Patara, S. Sampietro, and E. Vicario, “A framework for Model-Driven Engineering of resilient software-controlled systems,” *Computing*, vol. 103, no. 4, pp. 589–612, Apr. 2021.

[65] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, “Adaptive federated learning and digital twin for industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5605–5614, 2021.

[66] J. Kephart and D. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[67] J. Dobaj, A. Riel, T. Krug, M. Seidl, G. Macher, and M. Egretzberger, “Towards digital twin-enabled devops for cps providing architecture-based service adaptation & verification at runtime,” in *Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2022, p. 132–143.

[68] L. F. Rivera, H. A. Müller, N. M. Villegas, G. Tamura, and M. Jiménez, “On the engineering of iot-intensive digital twin software systems,” in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 2020, p. 631–638.

[69] E. Kamburjan, C. C. Din, R. Schlatte, S. L. T. Tarifa, and E. B. Johnsen, “Twinning-by-Construction: Ensuring Correctness for Self-adaptive Digital Twins,” in *Leveraging Applications of Formal Methods, Verification and Vali-*

dation. Verification Principles. Cham, Switzerland: Springer, Oct. 2022, pp. 188–204.

[70] F. Edrisi, D. Perez-Palacin, M. Caporuscio, and S. Giussani, "Adaptive controllers and digital twin for self-adaptive robotic manipulators," in *2023 IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2023, pp. 56–67.

[71] H. Van der Valk, H. Haße, F. Möller, M. Arbter, J.-L. Henning, and B. Otto, "A taxonomy of digital twins." in *Americas Conference on Information Systems*, 2020.

[72] C. Galera-Zarco and E. Papadonikolaki, "Digital twin-based services: a taxonomy," in *International Conference on AI and the Digital Economy (CADE 2023)*, vol. 2023, 2023, pp. 98–102.

[73] I. David and D. Bork, "Towards a taxonomy of digital twin evolution for technical sustainability," in *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2023, pp. 934–938.

[74] Y. Yang, Y. Shi, C. Yi, J. Cai, J. Kang, D. Niyato, and X. Shen, "Dynamic human digital twin deployment at the edge for task execution: A two-timescale accuracy-aware online optimization," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024.

[75] T. Clark, V. Kulkarni, J. Whittle, and R. Breu, "Engineering digital twin-enabled systems," *IEEE Software*, vol. 39, no. 2, pp. 16–19, 2022.

[76] G. Shao and M. Helu, "Framework for a digital twin in manufacturing: Scope and requirements," *Manufacturing letters*, vol. 24, pp. 105–107, 2020.

[77] J. Li, M. Liu, W. Wang, and C. Hu, "Inspection robot based on offline digital twin synchronization architecture," *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 943–947, 2022.

[78] J. Ahlgren, K. Bojarczuk, S. Drossopoulou, I. Dvortsova, J. George, N. Gucevska, M. Harman, M. Lomeli, S. M. M. Lucas, E. Meijer, S. Omohundro, R. Rojas, S. Sapora, and N. Zhou, "Facebook's Cyber–Cyber and Cyber–Physical Digital Twins," in *EASE 2021: Evaluation and Assessment in Software Engineering*, Jun 2021, pp. 1–9.

[79] R. van Dinter, B. Tekinerdogan, and C. Catal, "Predictive maintenance using digital twins: A systematic literature review," *Information and Software Technology*, vol. 151, p. 107008, Nov. 2022.

[80] B. S. Barn, "The sociotechnical digital twin: on the gap between social and technical feasibility," in *2022 IEEE 24th Conference on Business Informatics (CBI)*, vol. 1. IEEE, 2022, pp. 11–20.

[81] H. Zhang, G. Zhang, and Q. Yan, "Digital twin-driven cyber-physical production system towards smart shop-floor," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 11, pp. 4439–4453, 2019.

[82] F. Mostafa, L. Tao, and W. Yu, "An effective architecture of digital twin system to support human decision making and AI-driven autonomy," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 19, p. e6111, Oct. 2021.

[83] M. Waszak, A. N. Lam, V. Hoffmann, B. Elvesæter, M. F. Mogos, and D. Roman, "Let the asset decide: Digital twins with knowledge graphs," in *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*, 2022, pp. 35–39.

[84] C. Steinmetz, G. N. Schroeder, R. N. Rodrigues, A. Rettberg, and C. E. Pereira, “Key-components for digital twin modeling with granularity: Use case car-as-a-service,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 23–33, 2022.

[85] P. Franceschi, S. Mutti, K. Ottogalli, D. Rosquete, D. Borro, and N. Pedrocchi, “A framework for cyber-physical production system management and digital twin feedback monitoring for fast failure recovery,” *International Journal of Computer Integrated Manufacturing*, Jun. 2022.

[86] I. A. Tsokalo, D. Kuss, I. Kharabet, F. H. P. Fitzek, and M. Reisslein, “Remote robot control with human-in-the-loop over long distances using digital twins,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[87] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. H. L. Hansen, “Vision-cloud data fusion for adas: A lane change prediction case study,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 210–220, 2022.

[88] Z. Wang, X. Liao, C. Wang, D. Oswald, G. Wu, K. Boriboonsomsin, M. J. Barth, K. Han, B. Kim, and P. Tiwari, “Driver behavior modeling using game engine and real vehicle: A learning-based approach,” *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 738–749, 2020.

[89] M. Ronzoni, R. Accorsi, L. Botti, and R. Manzini, “A support-design framework for Cooperative Robots systems in labor-intensive manufacturing processes,” *Journal of Manufacturing Systems*, vol. 61, pp. 646–657, Oct. 2021.

[90] M. Peruzzini, F. Grandi, and M. Pellicciari, “Exploring the potential of Operator 4.0 interface and monitoring,” *Computers & Industrial Engineering*, vol. 139, p. 105600, Jan. 2020.

[91] I. A. Tsokalo, D. Kuss, I. Kharabet, F. H. Fitzek, and M. Reisslein, “Remote robot control with human-in-the-loop over long distances using digital twins,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[92] E. Yigitbas, K. Karakaya, I. Jovanovikj, and G. Engels, “Enhancing human-in-the-loop adaptive systems through digital twins and vr interfaces,” in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE, 2021, pp. 30–40.

[93] È. Pairet, P. Ardón, X. Liu, J. Lopes, H. Hastie, and K. S. Lohan, “A digital twin for human-robot interaction,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 372–372.

[94] B. A. Yilma, H. Panetto, and Y. Naudet, “Systemic formalisation of Cyber-Physical-Social System (CPSS): A systematic literature review,” *Computers in Industry*, vol. 129, p. 103458, Aug. 2021.

[95] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti, “Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review,” *Journal of manufacturing systems*, vol. 58, pp. 176–192, 2021.

[96] J. Huang, F. Bastani, I. . Yen, J. Dong, W. Zhang, F. . Wang, and H. . Hsu, “Extending service model to build an effective service composition framework for cyber-physical systems,” in *2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2009, pp. 1–8.

[97] T. Wang, L. Cheng, and K. Zhang, “An efficient context-sensitive service composition framework for precise controlling in cyber-physical systems,” in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 1, 2011, pp. 340–343.

[98] W. Zhu, G. Zhou, I. Yen, and F. Bastani, "A pt-soa model for cps/iot services," in *2015 IEEE International Conference on Web Services (ICWS)*, 2015, pp. 647–654.

[99] Y. Sun, G. Yang, and X. Zhou, "A novel ontology-based service model for cyber physical system," in *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, 2016, pp. 125–131.

[100] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, "Contextualizing ontologies," *Journal of Web Semantics*, vol. 1, no. 4, pp. 325–343, Oct. 2004.

[101] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere, "Towards an Extensible Context Ontology for Ambient Intelligence," in *Ambient Intelligence*. Springer, 2004, pp. 148–159.

[102] X. Wu, M. C. Sturman, and C. Wang, "The motivational effects of pay fairness: A longitudinal study in chinese star-level hotels," *Cornell Hospitality Quarterly*, vol. 54, no. 2, pp. 185–198, 2013.

[103] Z. Yu, H. Ma, B. Guo, and Z. Yang, "Crowdsensing 2.0," *Communications of the ACM*, vol. 64, no. 11, pp. 76–80, Oct. 2021.

[104] X. Zhang, Z. Yang, Y. Liu, and S. Tang, "On Reliable Task Assignment for Spatial Crowdsourcing," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 1, pp. 174–186, Sep. 2016.

[105] H. To, C. Shahabi, and L. Kazemi, "A Server-Assigned Spatial Crowdsourcing Framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, pp. 1–28, Jul. 2015.

[106] L. Tran, H. To, L. Fan, and C. Shahabi, "A real-time framework for task assignment in hyperlocal spatial crowdsourcing," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 3, jan 2018.

[107] F. Basık, B. Gedik, H. Ferhatosmanoğlu, and K.-L. Wu, "Fair task allocation in crowdsourced delivery," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1040–1053, 2018.

[108] Z. Chen, P. Cheng, L. Chen, X. Lin, and C. Shahabi, "Fair task assignment in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2479–2492, 2020.

[109] J. Lan, Y. Shao, X. Gao, and G. Chen, "Fatp: Fairness-aware task planning in spatial crowdsourcing," in *2020 IEEE International Conference on Web Services (ICWS)*. IEEE, 2020, pp. 256–264.

[110] Y. Zhao, K. Zheng, J. Guo, B. Yang, T. B. Pedersen, and C. S. Jensen, "Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 265–276.

[111] D. Weyns, N. Bencomo, R. Calinescu, J. Camara, C. Ghezzi, V. Grassi, L. Grunske, P. Inverardi, J.-M. Jezequel, S. Malek *et al.*, "Perpetual assurances for self-adaptive systems," in *Software Engineering for Self-Adaptive Systems III. Assurances: International Seminar, Dagstuhl Castle, Germany, December 15-19, 2013, Revised Selected and Invited Papers*. Springer, 2017, pp. 31–63.

[112] J. Xu, K. Chen, and S. Reiff-Marganiec, "Using markov decision process model with logic scoring of preference model to optimize htn web services composition," *International Journal of Web Services Research (IJWSR)*, vol. 8, no. 2, pp. 53–73, 2011.

[113] L. Ren, W. Wang, and H. Xu, “A reinforcement learning method for constraint-satisfied services composition,” *IEEE Transactions on Services Computing*, vol. 13, no. 5, pp. 786–800, 2020.

[114] C. Ghezzi, L. S. Pinto, P. Spoletini, and G. Tamburrelli, “Managing non-functional uncertainty via model-driven adaptivity,” in *2013 35th International Conference on Software Engineering*, 2013, pp. 33–42.

[115] H. Wang, X. Wang, and Q. Yu, “Optimal self-healing of service-oriented systems with incomplete information,” in *2013 IEEE International Congress on Big Data*, 2013, pp. 227–234.

[116] A. Pandey, G. A. Moreno, J. Cámara, and D. Garlan, “Hybrid planning for decision making in self-adaptive systems,” in *2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, 2016, pp. 130–139.

[117] G. A. Moreno, J. Cámara, D. Garlan, and B. Schmerl, “Proactive self-adaptation under uncertainty: a probabilistic model checking approach,” in *Proceedings of the 2015 10th joint meeting on foundations of software engineering*, 2015, pp. 1–12.

[118] ——, “Flexible and efficient decision-making for proactive latency-aware self-adaptation,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 13, no. 1, pp. 1–36, 2018.

[119] R. P. Goldman, D. J. Musliner, and K. D. Krebsbach, “Managing Online Self-adaptation in Real-Time Environments,” in *Self-Adaptive Software: Applications*. Berlin, Germany: Springer, Jun. 2003, pp. 6–23.

[120] M. Hamadouche, C. Dezan, and K. R. L. J. C. Branco, “Reward tuning for self-adaptive policy in mdp based distributed decision-making to ensure a safe

mission planning," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 78–85.

[121] J. Liang, M. Ma, and X. Tan, "Gadqn-ids: A novel self-adaptive ids for vanets based on bayesian game theory and deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 724–12 737, 2021.

[122] J. Cámara, W. Peng, D. Garlan, and B. Schmerl, "Reasoning about sensing uncertainty and its reduction in decision-making for self-adaptation," *Science of Computer Programming*, vol. 167, pp. 51–69, Dec. 2018.

[123] IBM, "What is observability?" Sep. 2023, [accessed 17. Sep. 2023]. [Online]. Available: https://www.ibm.com/topics/observability

[124] L. H. Garcia Paucar and N. Bencomo, "Re-storm: Mapping the decision-making problem and non-functional requirements trade-off to partially observable markov decision processes," in *2018 IEEE/ACM 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2018, pp. 19–25.

[125] L. H. G. Paucar and N. Bencomo, "Knowledge base k models to support trade-offs for self-adaptation using markov processes," in *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, 2019, pp. 11–16.

[126] H. Samin, N. Bencomo, and P. Sawyer, "Pri-aware: Tool support for priority-aware decision-making under uncertainty," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 450–451.

[127] M. Abundo, V. Cardellini, and F. L. Presti, "An mdp-based admission control for a qos-aware service-oriented system," in *2011 IEEE Nineteenth IEEE International Workshop on Quality of Service*, 2011, pp. 1–3.

[128] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[129] L. H. G. Paucar, N. Bencomo, and K. K. F. Yuen, "Arrow: automatic runtime reappraisal of weights for self-adaptation," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, p. 1584–1591.

[130] P. Sobalvarro, "Here's why human-robot collaboration is the future of manufacturing," Aug 2020, [accessed 25. Sept. 2020]. [Online]. Available: https://www.weforum.org/agenda/2020/08/here-s-how-robots-can-help-us-confront-covid

[131] T. Endsley and B. Appleby, "Taxonomy for applied human-machine teaming for space system domains," in *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–9.

[132] M. A. Boden, *The Creative Mind: Myths and Mechanisms.* London, England, UK: Abacus, Apr. 1992. [Online]. Available: https://www.amazon.co.uk/Creative-Mind-Myths-Mechanisms/dp/0349104697

[133] D. N. Crowley, E. Curry, and J. G. Breslin, "Closing the loop - from citizen sensing to citizen actuation," in *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2013, pp. 108–113.

[134] S. Wang, A. Zhou, M. Yang, L. Sun, C. Hsu, and F. Yang, "Service composition in cyber-physical-social systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 82–91, 2020.

[135] T. Wang, C. Niu, and L. Cheng, "A two-phase context-sensitive service composition method with the workflow model in cyber-physical systems," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014, pp. 1475–1482.

[136] A. V. Feljan, S. K. Mohalik, M. B. Jayaraman, and R. Badrinath, "Soa-pe: A service-oriented architecture for planning and execution in cyber-physical systems," in *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, 2015, pp. 1–6.

[137] A. Elhabbash, V. Nundloll, Y. Elkhatib, G. S. Blair, and V. S. Marco, "An ontological architecture for principled and automated system of systems composition," in *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2020, pp. 85–95.

[138] A. Elhabbash, Y. Elkhatib, V. Nundloll, V. S. Marco, and G. S. Blair, "Principled and automated system of systems composition using an ontological architecture," *Future Generation Computer Systems*, vol. 157, pp. 499–515, 2024.

[139] H. J. La and S. D. Kim, "A service-based approach to designing cyber physical systems," in *2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, 2010, pp. 895–900.

[140] K. Huang, J. Yao, J. Zhang, and Z. Feng, "Human-as-a-service: Growth in human service ecosystem," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 90–97.

[141] ——, "When human service meets crowdsourcing: Emerging in human service collaboration," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 460–473, 2018.

[142] A. Computing *et al.*, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, no. 2006, pp. 1–6, 2006.

[143] R. Kazhamiakin, S. Benbernou, L. Baresi, P. Plebani, M. Uhlig, and O. Barais, "Adaptation of service-based systems," in *Service research challenges and solutions for the future internet*. Springer, 2010, pp. 117–156.

[144] "Standard Occupational Classification (SOC) - Office for National Statistics," Nov 2020, [accessed 11. Nov. 2020]. [Online]. Available: https://www.ons.gov.uk/methodology/classificationsandstandards/standardoccupationalclassificationsoc

[145] "ISCO - International Standard Classification of Occupations," 2008, [accessed 10. Nov. 2020]. [Online]. Available: https://www.ilo.org/public/english/bureau/stat/isco

[146] L. Klinestiver, "Psychophysiological and other factors affecting human performance in accident prevention and investigation," EG and G Idaho, Tech. Rep., 1980.

[147] R. A. Bates, T. Hatcher, E. F. Holton III, and N. Chalofsky, "Redefining human resource development: An integration of the learning, performance, and spirituality of work perspectives," in *Academy of Human Resource Development Conference Proceedings. Baton Rouge: AHRD*, 2001.

[148] P. Paleri, "Active human to maximum human," in *Human Investment Management*. Springer, 2018, pp. 81–111.

[149] C. P. Vogt, "Maximizing human potential: Capabilities theory and the professional work environment," *Journal of Business Ethics*, vol. 58, no. 1-3, pp. 111–123, 2005.

[150] D. Weyns, B. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, and K. M. Göschka, "On patterns for decentralized control in self-adaptive systems," in *Software Engineering for Self-Adaptive Systems II*. Springer, 2013, pp. 76–107.

[151] D. Vrandečić, "Ontology evaluation," in *Handbook on ontologies*. Springer, 2009, pp. 293–313.

[152] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, “Ontology evaluation and validation: an integrated formal model for the quality diagnostic task,” Laboratory of Applied Ontologies – CNR, Rome, Italy, Tech. Rep., 2005, http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf.

[153] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, “Hermit: an owl 2 reasoner,” *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.

[154] N. Guarino and C. A. Welty, “An Overview of OntoClean,” in *Handbook on Ontologies*. Berlin, Germany: Springer, 2004, pp. 151–171.

[155] Y. Song, F. R. Yu, L. Zhou, X. Yang, and Z. He, “Applications of the Internet of Things (IoT) in Smart Logistics: A Comprehensive Survey,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4250–4274, Oct. 2020.

[156] A. Kshirsagar, B. Dreyfuss, G. Ishai, O. Heffetz, and G. Hoffman, “Monetary-Incentive Competition Between Humans and Robots: Experimental Results,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Mar. 2019, pp. 95–103.

[157] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, “Extortion and cooperation in rating protocol design for competitive crowdsourcing,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 246–259, 2021.

[158] E. Fehr and K. M. Schmidt, “A theory of fairness, competition, and cooperation,” *The quarterly journal of economics*, vol. 114, no. 3, pp. 817–868, 1999.

[159] A. M. Brawley and C. L. Pury, “Work experiences on mturk: Job satisfaction, turnover, and information sharing,” *Computers in Human Behavior*, vol. 54, pp. 531–546, 2016.

[160] J. M. Zhang and M. Harman, “Ignorance and prejudice in software fairness,” in *2021 IEEE/ACM 43rd International Conference on Software Engineering*, 2021, pp. 1436–1447.

[161] A. Kaufman, “Distributive Justice, Theories of,” in *Encyclopedia of Applied Ethics (Second Edition)*. Cambridge, MA, USA: Academic Press, Jan. 2012, pp. 842–850.

[162] M. Deutsch, “Equity, equality, and need: What determines which value will be used as the basis of distributive justice?” *Journal of Social issues*, 1975.

[163] R. Folger, B. H. Sheppard, and R. T. Buttram, “Equity, equality, and need: Three faces of social justice.” in *Conflict, cooperation, and justice: Essays inspired by the work of Morton Deutsch*, B. B. Bunker and J. Z. Rubin, Eds. Jossey-Bass/Wiley, 1995, pp. 261–289.

[164] G. F. Wagstaff, “Equity, equality, and need: Three principles of justice or one? an analysis of “equity as desert”,” *Current psychology*, vol. 13, no. 2, pp. 138–152, 1994.

[165] H.-W. Bierhoff and E. Rohmann, “Justice in performance situations: Compromise between equity and equality,” in *Justice and Conflicts*. Springer, 2011, pp. 135–152.

[166] J. Salminen, A. M. S. Kamel, S.-G. Jung, M. Mustak, and B. J. Jansen, “Fair compensation of crowdsourcing work: the problem of flat rates,” *Behaviour & Information Technology*, pp. 1–22, Nov. 2022.

[167] D. R. Forsyth, *Group Dynamics*, 5th ed. Australia ; Belmont, CA: Wadsworth, Cengage Learning, 2010.

[168] T. Schwinger, “The need principle of distributive justice,” in *Justice in social relations*. Springer, 1986, pp. 211–225.

[169] W. Miao, Y. Deng, W. Wang, Y. Liu, and C. S. Tang, “The effects of surge pricing on driver behavior in the ride-sharing market: Evidence from a quasi-experiment,” *Journal of Operations Management*, vol. 69, no. 5, pp. 794–822, Jul. 2023.

[170] M. Kavaliova, F. Virjee, N. Maehle, and I. A. Kleppe, “Crowdsourcing innovation and product development: Gamification as a motivational driver,” *Cogent Business & Management*, vol. 3, no. 1, p. 1128132, 2016.

[171] R. R. Schutz, “On the measurement of income inequality,” *The American Economic Review*, vol. 41, no. 1, pp. 107–122, 1951.

[172] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, “A quantitative measure of fairness and discrimination,” *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, vol. 21, 1984.

[173] R. Hamadi, H. Ghazzai, H. Besbes, and Y. Massoud, “Financial advisor recruitment: A smart crowdsourcing-assisted approach,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 682–688, 2021.

[174] R. Duan and H.-H. Su, “A scaling algorithm for maximum weight matching in bipartite graphs,” in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, 2012, pp. 1413–1424.

[175] B. Huang, H. Zhu, D. Liu, N. Wu, Y. Qiao, and Q. Jiang, “Solving last-mile logistics problem in spatiotemporal crowdsourcing via role awareness with adaptive clustering,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 668–681, 2021.

[176] W. Dai, Y. Wang, J. Ma, and Q. Jin, “Btr: A feature-based bayesian task recommendation scheme for crowdsourcing system,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 780–789, 2020.

[177] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *VLDB Journal*, vol. 29, no. 1, pp. 217–250, Jan. 2020.

[178] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: experiments and analysis," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1053–1064, 2016.

[179] L. Kazemi and C. Shahabi, "GeoCrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, Nov. 2012, pp. 189–198.

[180] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, 2008, pp. 11–15.

[181] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[182] C. Calhoun, *Dictionary of the Social Sciences.* Oxford University Press, 2002. [Online]. Available: https://www.oxfordreference.com/view/10.1093/acref/9780195123715.001.0001/acref-9780195123715

[183] S. Hawking and L. Mlodinow, *The Grand Design*, Sep. 2010. [Online]. Available: https://www.goodreads.com/book/show/8520362-the-grand-design

[184] M. Salama, R. Bahsoon, and N. Bencomo, "Managing trade-offs in self-adaptive software architectures: A systematic mapping study," *Managing trade-offs in adaptable software architectures*, pp. 249–297, 2017.

[185] T. Dürschmid, E. Kang, and D. Garlan, "Trade-off-oriented development: making quality attribute trade-offs first-class," in *2019 IEEE/ACM 41st In-*

ternational Conference on Software Engineering: New Ideas and Emerging Results. IEEE, 2019, pp. 109–112.

[186] D. Leffingwell and S. A. Team, "Nonfunctional Requirements - Scaled Agile Framework," Apr. 2023, [accessed 22. Jun. 2023]. [Online]. Available: https://scaledagileframework.com/nonfunctional-requirements

[187] M. Ji, A. Veitch, and J. Wilkes, "Seneca: Remote mirroring done write," in *2003 USENIX Annual Technical Conference (USENIX ATC 03)*. San Antonio, TX: USENIX Association, Jun. 2003.

[188] H. Samin, L. H. G. Paucar, N. Bencomo, C. M. C. Hurtado, and E. M. Fredericks, "Rdmsim: An exemplar for evaluation and comparison of decision-making techniques for self-adaptation," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2021, pp. 238–244.

[189] D. M. Roijers, S. Whiteson, and F. A. Oliehoek, "Point-based planning for multi-objective pomdps," in *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI)*, vol. 2015. AAAI Press, 2015, pp. 1666–1672.

[190] P. Emami, A. J. Hamlet, and C. D. Crane, "Pomdpy: An extensible framework for implementing partially-observable markov decision processes in python," 2015. [Online]. Available: https://pemami4911.github.io/pdfs/pomdpy-extensible-framework.pdf

[191] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[192] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[193] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues, "Model-based quantitative network security metrics: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2704–2734, 2017.

[194] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[195] C. Werner, Z. S. Li, D. Lowlind, O. Elazhary, N. Ernst, and D. Damian, "Continuously managing nfrs: Opportunities and challenges in practice," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2629–2642, 2022.

[196] C. Boutilier, "A pomdp formulation of preference elicitation problems," in *Eighteenth National Conference on Artificial Intelligence.* American Association for Artificial Intelligence, 2002, p. 239–246.

[197] K. Wiegers and J. Beatty, *Software requirements.* Pearson Education, 2013.

[198] I. Ognjanovic, D. Gašević, E. Bagheri, and M. Asadi, "Conditional preferences in software stakeholders' judgments," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, p. 683–690.

[199] N. Kukreja, "Decision theoretic requirements prioritization: A two-step approach for sliding towards value realization," in *Proceedings of the 2013 International Conference on Software Engineering*, 2013, p. 1465–1467.

[200] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb, "Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead," *Management science*, vol. 54, no. 7, pp. 1336–1349, 2008.

[201] E. Karni and D. Schmeidler, "Utility theory with uncertainty," *Handbook of mathematical economics*, vol. 4, pp. 1763–1831, 1991.

[202] J. Whittle, P. Sawyer, N. Bencomo, B. H. Cheng, and J.-M. Bruel, "Relax: Incorporating uncertainty into the specification of self-adaptive systems," in *2009 17th IEEE International Requirements Engineering Conference*, 2009, pp. 79–88.

[203] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature-PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings 8*. Springer, 2004, pp. 722–731.

[204] H. T. N. Ignatius, "XPOMDPy," Sep. 2023. [Online]. Available: https://github.com/hxi903/xpomdpy

[205] E. Walraven and M. Spaan, "Accelerated vector pruning for optimal pomdp solvers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[206] L. N. Alegre, A. Bazzan, and B. C. Da Silva, "Optimistic linear support and successor features as a basis for optimal policy transfer," in *International Conference on Machine Learning*. PMLR, 2022, pp. 394–413.

[207] A. Behl, G. A. Wainer, and C. Ruiz-Martin, "Cell-devs: an approach to model the influence of social interactions in human behavior," in *Proceedings of the 50th Computer Simulation Conference*, ser. SummerSim '18. San Diego, CA, USA: Society for Computer Simulation International, 2018.

[208] N. Noceti and F. Odone, "Humans in groups: The importance of contextual information for understanding collective activities," *Pattern Recognition*, vol. 47, no. 11, pp. 3535–3551, Nov. 2014.

[209] J. Kiseleva, "Using Contextual Information to Understand Searching and Browsing Behavior," in *SIGIR '15: Proceedings of the 38th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, Aug. 2015, p. 1059.

[210] J. Ye, H. Hu, G.-J. Qi, and K. A. Hua, "A temporal order modeling approach to human action recognition from multimodal sensor data," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, pp. 1–22, 2017.

[211] Z. Sunberg and M. J. Kochenderfer, "Improving automated driving through pomdp planning with human internal states," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20 073–20 083, 2022.

[212] T. F. Monaghan, S. N. Rahman, C. W. Agudelo, A. J. Wein, J. M. Lazar, K. Everaert, and R. R. Dmochowski, "Foundational statistical principles in medical research: sensitivity, specificity, positive predictive value, and negative predictive value," *Medicina*, vol. 57, no. 5, p. 503, 2021.

[213] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces," 2009.

[214] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, 2017.

[215] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Reinforcement learning for non-stationary Markov decision processes: The blessing of (More) optimism," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1843–1854.

[216] T. Dick, A. Gyorgy, and C. Szepesvari, "Online learning in markov decision processes with changing cost sequences," in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1, Bejing, China, 22–24 Jun 2014, pp. 512–520.

[217] Z. Wang, X. Liao, C. Wang, D. Oswald, G. Wu, K. Boriboonsomsin, M. J. Barth, K. Han, B. Kim, and P. Tiwari, "Driver behavior modeling using game engine and real vehicle: A learning-based approach," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 738–749, 2020.

[218] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 198–209, 2021.

[219] K. Kuru, "Conceptualisation of human-on-the-loop haptic teleoperation with fully autonomous self-driving vehicles in the urban environment," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 448–469, 2021.

[220] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. H. L. Hansen, "Vision-cloud data fusion for adas: A lane change prediction case study," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 210–220, 2022.

[221] M. Fennel, A. Zea, J. Mangler, A. Roennau, and U. D. Hanebeck, "Haptic rendering of arbitrary serial manipulators for robot programming," *IEEE Control Systems Letters*, vol. 6, pp. 716–721, 2022.

[222] Z. Xu, T. Jiang, and N. Zheng, "Developing and analyzing eco-driving strategies for on-road emission reduction in urban transport systems - A VR-enabled digital-twin approach," *Chemosphere*, vol. 305, p. 135372, Oct. 2022.

[223] S. Koukas, N. Kousi, S. Aivaliotis, G. Michalos, R. Bröchler, and S. Makris, "ODIN architecture enabling reconfigurable human – robot based production lines," *Procedia CIRP*, vol. 107, pp. 1403–1408, Jan. 2022.

[224] P. Kunze, S. Rein, M. Hemsendorf, K. Ramspeck, and M. Demant, "Learning an Empirical Digital Twin from Measurement Images for a Comprehensive Quality Inspection of Solar Cells," *Solar RRL*, vol. 6, no. 5, p. 2100483, May 2022.

[225] X. Wang, C.-J. Liang, C. C. Menassa, and V. R. Kamat, “Interactive and immersive process-level digital twin for collaborative human–robot construction work,” *Journal of Computing in Civil Engineering*, vol. 35, no. 6, p. 04021023, 2021.

[226] C. Li, P. Zheng, S. Li, Y. Pang, and C. K. Lee, “Ar-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop,” *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102321, 2022.

Appendix A

Classification of Digital Twin Applications in HitLCPS Based on Taxonomy

We take a sample of 15 HitLCPS articles that apply digital twins in various domains, from transportation to manufacturing. We classify these studies using the taxonomy we proposed in Chapter 2, which we present in Tables A.1 and A.2.

Table A.1: Classifying digital twins by human modes, temporal integration, deployment, and maturity level

No	Article	Human modes		Temporal Integration		Deployment		Maturity Level				
		Off-twin	On-twin	Offline	Online	Design time	Runtime	Digital Blueprint	Digital Model	Digital Shadow	Pure-play	Assisted Pure-play
1	Pairet et al. [93]	x		x		x		x				
2	Yigitbas et al. [92]	x			x		x					x
3	Tsokalo et al. [86]	x			x		x					x
4	Wang et al. [217]		x	x		x		x				
5	Wang et al. [218]		x		x		x				x	
6	K. Kuru [219]	x			x		x					x
7	Liu et al. [220]		x		x		x					x
8	Fennel et al. [221]	x		x		x			x			
9	Xu et al. [222]		x	x		x			x			
10	Koukas et al. [223]		x		x		x				x	
11	Ronzoni et al. [89]		x	x		x		x				
12	Li et al. li2022ar	x			x		x				x	
13	Kunze et al. [224]	x			x	x		x				
14	Franceschi et al. [85]	x			x		x			x		
15	Wang et al. [225]	x			x		x					x

Table A.2: Classifying digital twins by twinning type, role of human, and feedback.

No	Article	Twinning Type		Role of human		Feedback			
		Cyber-Cyber	Cyber-Physical	Service Consumer	Service Provider	Descriptive	Diagnostic	Prognostic	Prescriptive
1	Pairet et al. [93]		x	x		x			
2	Yigitbas et al. [92]		x	x		x	x		x
3	Tsokalo et al. [86]		x	x		x			x
4	Wang et al. [217]		x	x				x	x
5	Wang et al. [218]	x		x				x	x
6	K. Kuru [219]		x	x	x			x	x
7	Liu et al. [220]	x		x	x			x	x
8	Fennel et al. [221]		x	x					x
9	Xu et al. [222]		x	x		x			x
10	Koukas et al. [223]		x	x	x				x
11	Ronzoni et al. [89]		x	x	x	x			
12	Li et al. [226]		x	x		x			x
13	Kunze et al. [224]	x			x	x			
14	Franceschi et al. [85]		x	x	x	x	x		
15	Wang et al. [225]	x		x		x			