Real-time Optimization of Context-Aware Adaptive User Interfaces, for Enhanced Situational Awareness

Zinovia Stefanidi

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, Heraklion, Crete, GR-70013, Greece

Thesis Advisor: Associate Professor *George Papagiannakis*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

UNIVERSITY OF CRETE
COMPUTER SCIENCE DEPARTMENT

Real-time Optimization of Context-Aware Adaptive User Interfaces, for Enhanced Situational Awareness

Thesis submitted by
Zinovia Stefanidi
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: Zinovia Stefanidi

Committee approvals:

GEORGIOS PAPAGIANNAKIS Digitally signed by GEORGIOS PAPAGIANNAKIS Date: 2021.10.19 15:05:15 +03'00'

George Papagiannakis
Associate Professor, Thesis Supervisor

Panagiotis Tsakalidis Digitally signed by Panagiotis Tsakalidis Date: 2021.10.19 15:32:23 +03'00'

Panagiotis Tsakalides
Professor, Committee Member

XENOFON ZAMPOULIS Digitally signed by XENOFON ZAMPOULIS Date: 2021.10.19 15:48:31 +03'00'

Dr. Xenophon Zabulis
Research Director, ICS-FORTH, Committee Member

Departmental approval:

Polyvios Pratikakis Digitally signed by Polyvios Pratikakis Date: 2021.10.20 15:11:59 +03'00'

Polyvios Pratikakis
Associate Professor, Director of Graduate Studies

Heraklion, October 2021

Real-time Optimization of Context-Aware Adaptive User Interfaces, for Enhanced Situational Awareness

Abstract

User Interfaces (UIs) constitute the prominent means for interacting with computing systems and applications. Designing suitable, user-friendly UIs poses a multitude of challenges, given the heterogeneity of potential users and contexts of use. This variability cannot be handled by a one-size-fits-all approach, but needs to be addressed by adapting the UI so that it is tailored to the current user and context. Existing approaches are mainly focused on design-time or one-off adaptation of the UI at startup, as opposed to real-time continuous adaptation based on the current situation. However, UIs are nowadays increasingly being used in continuously changing contexts, such as in mobile and Extended Reality (XR) applications, calling for more dynamic approaches.

The majority of research approaches regarding adaptive Graphical User Interfaces (GUIs) is primarily concerned with the development of handcrafted rule sets and heuristics. Albeit in recent years, Combinatorial Optimization has emerged as a powerful and flexible tool for the computational generation and adaptation of GUIs, providing a coherent formalism for expressing and analyzing design decisions. In general, this method treats interface adaptation and generation as an optimization problem, by defining constraints and maximizing (or minimizing) an objective function that represents the goal of the UI, for instance, maximizing the interface's usability, or minimizing user effort. However, in existing approaches, the parameters of the optimization problem are manually specified or static, and do not reflect run-time changes in the current context of use. In addition, different types of design problems in a given UI, such as the selection of its GUI elements and its layout, are solved separately and independently. Finally, the layout of the UIs is either fixed at design-time, or it consists of predefined positions, regardless of the scene in the user's current field of view.

A key UI design consideration in many application domains, such as healthcare, aviation and the military, is Situational Awareness (SA), playing a major role in risk management and safety. It refers to the human perception and understanding of the environment and the current situation, as well as the human ability to predict how they will evolve. In this work, a novel computational approach for the dynamic adaptation of UIs is proposed, which aims at enhancing the SA of users by leveraging the current context and providing the most useful information, in an optimal and efficient manner. By combining Ontology modeling and reasoning with Combinatorial Optimization, the system decides *what* information to present, *when* to present it, *where* to visualize it in the display - and *how*, taking into consideration contextual factors as well as placement constraints. The main objective of the proposed approach is to optimize the SA associated with the displayed UI *at run-time*, while avoiding information overload and induced stress. In this respect, contrary to existing approaches, parameters of the optimization problem

are dynamically inferred, based on the current situation. Additionally, the design problems of GUI element selection and UI layout are solved simultaneously, exploiting interrelationships. Finally, the layout is dynamic, reflecting changes in the current scene.

Our proposed methodology is general-purpose, applicable to different platforms and domains, including desktop, mobile and XR applications, for a variety of potential end-users. In the context of this work, we have deployed our computational approach to the use case of an Augmented Reality (AR) system for Law Enforcement Agents (LEAs). In order to extract user requirements and model our application domain, co-creation workshops with end-users have been organized, gaining insights into context factors that impact the SA of LEAs, and identifying GUI elements that would increase their SA during policing in different tasks and contexts. To explore the benefits and limitations of the developed system, two evaluations have been conducted. The first one was an expert-based evaluation with LEAs and User Experience (UX) experts, assessing the appropriateness of the system's decisions. The second one was a user-based evaluation involving LEAs from different agencies, estimating the SA, the mental workload and the overall UX associated with the system, through an AR simulation. The results indicate that the system enhances SA, and while not imposing workload, it provides an overall positive UX. In particular, observed and perceived user SA is improved, by 9.25% and 25.63% respectively.

Βελτιστοποίηση, σε πραγματικό χρόνο, Προσαρμοστικών Διεπαφών Χρήστη που λαμβάνουν υπόψιν το Πλαίσιο Χρήσης, για επαυξημένη Επίγνωση της Κατάστασης

Περίληψη

Οι Διεπαφές Χρήστη (User Interfaces – UIs, εφεξής ΔΧ) αποτελούν το σημαντικότερο μέσο αλληλεπίδρασης με υπολογιστικά συστήματα και εφαρμογές. Ο σχεδιασμός κατάλληλων, φιλικών προς τον χρήστη ΔΧ δημιουργεί πολλές προκλήσεις, δεδομένης της ετερογένειας των δυνητικών χρηστών και του πλαισίου χρήσης. Αυτή η ποικιλία δεν μπορεί να αντιμετωπιστεί με μια ενιαία προσέγγιση, αλλά μέσω προσαρμογής της ΔΧ έτσι ώστε να εξατομικευτεί για τον τρέχοντα χρήστη και το τρέχον πλαίσιο χρήσης. Υφιστάμενες προσεγγίσεις επικεντρώνονται κυρίως στην προσαρμογή της ΔΧ κατά τον σχεδιασμό, ή εφάπαξ κατά την εκκίνηση της αλληλεπίδρασης, σε αντίθεση με τη συνεχή προσαρμογή της ΔΧ, σε πραγματικό χρόνο, με βάση την τρέχουσα κατάσταση. Ωστόσο, οι ΔΧ χρησιμοποιούνται στις μέρες μας όλο και περισσότερο σε διαρκώς μεταβαλλόμενες συνθήκες, όπως για παράδειγμα σε εφαρμογές για κινητές συσκευές και Εκτεταμένη Πραγματικότητα (XR), απαιτώντας πιο δυναμικές προσεγγίσεις.

Η πλειοψηφία των ερευνητικών προσεγγίσεων σχετικά με προσαρμοστικά (adaptive) γραφικά περιβάλλοντα Διεπαφών Χρήστη (Graphical User Interfaces - GUIs) αφορά κυρίως στη «χειρωνακτική» ανάπτυξη κανόνων και ευρετικών τεχνικών. Ωστόσο, τα τελευταία χρόνια, η Συνδυαστική Βελτιστοποίηση έχει αναδειχθεί ως ένα ισχυρό και ευέλικτο εργαλείο για την υπολογιστική παραγωγή και προσαρμογή γραφικών ΔΧ, παρέχοντας έναν συνεκτικό φορμαλισμό για τη διαμόρφωση και την ανάλυση σχεδιαστικών αποφάσεων. Γενικά, η μέθοδος αυτή αντιμετωπίζει την προσαρμογή και τη δημιουργία της ΔΧ ως πρόβλημα βελτιστοποίησης, καθορίζοντας περιορισμούς και μεγιστοποιώντας (ή ελαχιστοποιώντας) μια αντικειμενική συνάρτηση που αντιπροσωπεύει τον στόχο της ΔΧ, για παράδειγμα, μεγιστοποιώντας τη χρησιμότητα της ΔΧ, ή ελαχιστοποιώντας την «προσπάθεια» του χρήστη. Παρόλα αυτά, στις υφιστάμενες προσεγγίσεις, οι παράμετροι του προβλήματος βελτιστοποίησης καθορίζονται με «χειρωνακτικό» τρόπο ή είναι στατικές και δεν αντικατοπτρίζουν τις μεταβολές που επισυμβαίνουν σε πραγματικό χρόνο στο τρέχον πλαίσιο ρήσης. Επιπλέον, διαφορετικοί τύποι προβλημάτων σχεδιασμού μιας δεδομένης ΔΧ, όπως η επιλογή των στοιχείων μιας γραφικής ΔΧ και η διάταξή τους, λύνονται ξεχωριστά και ανεξάρτητα. Τέλος, η διάταξη της ΔΧ είτε καθορίζεται εξ αρχής, κατά τον σχεδιασμό, είτε αποτελείται από προκαθορισμένες θέσεις, ανεξάρτητες από τη σκηνή στο τρέχον οπτικό πεδίο του χρήστη.

Ένα βασικό στοιχείο που πρέπει να ληφθεί υπόψιν κατά το σχεδιασμό μιας ΔΧ σε πολλούς τομείς εφαρμογών, όπως η υγειονομική περίθαλψη, η αεροπλοΐα και οι ένοπλες δυνάμεις, είναι η Επίγνωση της Κατάστασης (ΕΚ), η οποία παίζει σημαντικό ρόλο στη διαχείριση των κινδύνων και στην ασφάλεια. Αναφέρεται στην ανθρώπινη αντίληψη και κατανόηση του περιβάλλοντος και της τρέχουσας κατάστασης, καθώς και στην

ικανότητα πρόβλεψης του πώς θα εξελιχθούν. Στο πλαίσιο αυτής της εργασίας, προτείνεται μια καινοτόμα υπολογιστική προσέγγιση για τη δυναμική προσαρμογή των ΔΧ, η οποία στοχεύει στην ενίσχυση της ΕΚ των χρηστών, αξιοποιώντας τη γνώση των τρεχουσών συνθηκών και παρέχοντας τις πιο χρήσιμες (κατάλληλες) πληροφορίες, με βέλτιστο και αποτελεσματικό τρόπο. Συνδυάζοντας την Οντολογική μοντελοποίηση και συλλογιστική με τη Συνδυαστική Βελτιστοποίηση, το σύστημα αποφασίζει *ποιές* πληροφορίες να παρουσιάσει, *πότε* να τις παρουσιάσει, *πού* να τις απεικονίσει στην οθόνη και *πώς*, λαμβάνοντας υπόψιν παράγοντες που αφορούν τις συνθήκες χρήσης καθώς και τους περιορισμούς τοποθέτησης στην οθόνη παρουσίασης. Ο κύριος στόχος της προτεινόμενης προσέγγισης είναι να βελτιστοποιηθεί η ΕΚ που σχετίζεται με την απεικονιζόμενη ΔΧ *κατά την εκτέλεση*, αποφεύγοντας παράλληλα την υπερφόρτωση του χρήστη με πληροφορίες και το συναφώς προκαλούμενο άγχος. Στο πλαίσιο αυτό, αντίθετα με υφιστάμενες προσεγγίσεις, παράμετροι του προβλήματος βελτιστοποίησης συνάγονται δυναμικά, βάση της τρέχουσας κατάστασης. Επιπρόσθετα, τα σχεδιαστικά προβλήματα της επιλογής των στοιχείων μιας γραφικής ΔΧ και της διάταξής τους λύνονται ταυτόχρονα, αξιοποιώντας αλληλοσυσχετίσεις.Τέλος, η διάταξη της ΔΧ είναι δυναμική, αντικατοπτρίζοντας τις αλλαγές που διαδραματίζονται στην τρέχουσα σκηνή.

Η προτεινόμενη μεθοδολογία είναι γενικής χρήσης, εφαρμόσιμη σε διαφορετικές πλατφόρμες και τομείς, συμπεριλαμβανομένων επιτραπέζιων, κινητών και Εκτεταμένης Πραγματικότητας εφαρμογών, για ποικίλους δυνητικούς τελικούς χρήστες. Στο πλαίσιο αυτής της εργασίας, εφαρμόσαμε την προτεινόμενη υπολογιστική προσέγγιση στην περίπτωση χρήσης ενός συστήματος Επαυξημένης Πραγματικότητας για Αστυνομικούς. Προκειμένου να εξαχθούν οι απαιτήσεις των χρηστών και να μοντελοποιηθεί ο συγκεκριμένος τομέας εφαρμογής, οργανώθηκαν σεμινάρια συνδημιουργίας με τελικούς χρήστες, που έδωσαν τη δυνατότητα της απόκτησης πληροφοριών για παράγοντες του πλαισίου χρήσης που επηρεάζουν την ΕΚ των Αστυνομικών, καθώς και του προσδιορισμού των γραφικών στοιχείων της ΔΧ που αυξάνουν την ΕΚ κατά τη διάρκεια της αστυνόμευσης σε διαφορετικές εργασίες και για διαφορετικές συνθήκες χρήσης. Για να διερευνηθούν τα οφέλη και οι περιορισμοί του συστήματος που αναπτύχθηκε, πραγματοποιήθηκαν δύο αξιολογήσεις. Η πρώτη ήταν μια εμπειρική αξιολόγηση με εμπειρογνώμονες Αστυνομικούς καθώς και ειδικούς σε θέματα Εμπειρίας Χρήστη (User Experience – UX), οι οποίοι αξιολόγησαν την καταλληλότητα των αποφάσεων του συστήματος. Η δεύτερη ήταν μια αξιολόγηση με τελικούς χρήστες που περιελάμβανε Αστυνομικούς από διαφορετικούς οργανισμούς, και αποσκοπούσε στην εκτίμηση της ΕΚ, του νοητικού φόρτου εργασίας και της γενικής Εμπειρίας Χρήστη αναφορικά με το σύστημα, μέσω μιας προσομοίωσης Επαυξημένης Πραγματικότητας. Τα αποτελέσματα υποδεικνύουν ότι το σύστημα ενισχύει την Επίγνωση της Κατάστασης (ΕΚ) και ενώ δεν επιβάλλει φόρτο εργασίας, παρέχει συνολικά θετική Εμπειρία Χρήστη. Συγκεκριμένα, η παρατηρούμενη και αντιληπτή ΕΚ χρήστη βελτιώνονται, κατά 9,25 % και 25,63 % αντίστοιχα.

Acknowledgements

I would first like to wholeheartedly thank my Thesis supervisor, Associate Professor George Papagiannakis, for his continuous support and trust in me. I am indebted to George Margetis and Stavroula Ntoa, my Thesis co-advisors, for their guidance and support throughout the course of this work. My appreciation is extended to Natasa Ntagianta, Irene Sikianaki and Katerina Valakou for their assistance in the user evaluation study. A separate mention of appreciation is due to Konstantinos Tzevelekakis, classmate and colleague, for his contribution to rendering the work on 'stress' less stressful.

Last but not least, I express my gratitude to my family for their eternal love and support.

I would like to acknowledge the support of the Institute of Computer Science of FORTH during my undergraduate studies (through the 'Stelios Orphanoudakis distinguished scholarship'), as well as throughout my postgraduate studies.

Part of the work reported in this Thesis was carried out in connection with the European Union's Horizon 2020 research and innovation program, under Grant Agreement No 883297 (project DARLENE: "Deep AR Law Enforcement Ecosystem"), in the context of which the pertinent user studies have been conducted following approval by the Social and Societal Ethics Committee of the Katholieke Universiteit Leuven (KUL approval number G-2021 09 2072).

to my family

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The User Interface (UI) constitutes an integral component of interactive systems and applications, with a decisive impact in their utility, accessibility and the overall User Experience (UX). Designing a suitable, user-friendly UI poses many challenges, given the heterogeneity of the possible contexts of use. This heterogeneity stems from the multiplicity and diversity of end users, environmental conditions and platforms. This context variability cannot be sufficiently addressed by constructing a variety of UIs for the same functionality, as this would lead to an exponentially large number of potential adaptations. At the same time, it cannot be handled by a one-size-fits-all design. Hence, it needs to be addressed by adapting the UI so that it is tailored to the current user and the surrounding situation.

This concept of extracting information from the environment and reacting to the changing requirements of use has been coined in the literature as "Context-Awareness" [133]. It was first introduced in the domain of ubiquitous computing, and has since rapidly expanded to other research areas, including Intelligent User Interfaces (IUIs) and Extended Reality (XR) applications. In order to capture and utilize the different properties and characteristics of contextual information, appropriate representation of, and reasoning about, context is a requisite. To this end, a multitude of modeling techniques and inference mechanisms have been proposed, with Ontology-based modeling being a powerful, widely adopted approach, supporting both representation and reasoning, and exhibiting clear benefits over competing approaches [144, 23, 157].

The power of Context-Awareness can be harnessed in a wide spectrum of application domains and for a multitude of purposes, including the adaptation of User Interfaces, relevant in the context of this work. Adaptive User Interfaces (AUIs) aim to suit the user's profile, preferences, interaction platform and computing environment, by appropriately modifying their content, presentation, as well as their input and output modalities [142]. Existing approaches are mainly focused on design-time or one-off adaptation of the UI at startup, as opposed to real-time continuous adaptation based on the current situation. However, UIs are nowadays increasingly being used in constantly changing contexts, such as in mobile and

Extended Reality (XR) applications, calling for more dynamic approaches.

Regarding adaptation techniques, the majority of research in AUIs is primarily concerned with the development of handcrafted rule sets and heuristics [89]. The creation of rules is carried out either with the help of UX experts, or system designers, or automatically deduced by user interaction data with the system. An emerging, potent technique for the automatic generation and adaptation of UIs, is Combinatorial Optimization (CO), solving design and content decisions as an optimization problem. To this end, an objective function is defined which expresses the aim of the UI: e.g. the maximization of the interface's usability [106], or the minimization of the user effort or selection time [18]. Appropriate constrains are also defined, for instance, with respect to users' capabilities [60] and capacities [106]. This flexible methodology allows identifying existing problem classes, assessing their complexity, and utilizing known algorithmic solutions. However, in existing approaches, the parameters of the optimization problem are manually specified or static. In particular, the "profit" or "cost" of individual UI decisions, commonly expressed as coefficients in the objective function [123], are defined a priori and do not reflect the variable and dynamic context in which the goal of the UI needs to be optimized, in many applications. Moreover, different types of design problems in a given UI, such as the selection of its GUI elements and its layout, are solved separately and independently, ignoring interrelations. Finally, the layout of the UIs is either optimized once, at design-time, or it consists of predefined, independent positions, regardless of what is currently happening in the scene, in the user's field of view.

A prime UI goal in a multitude of application domains, including healthcare, maintenance, mining, aviation and the military is Situational Awareness (SA) [49]. It is formally defined by Endsley et al. as *"the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future"* [46]. In particular, the theoretical model of SA [45] involves 3 levels: perceiving critical factors in the scene (Level 1 SA), understanding their meaning (Level 2 SA), and predicting how they will evolve (Level 3 SA). There exist numerous factors that can "hurt" SA - the so called "SA Daemons" [47]. Prominent such factors include stress, anxiety and workload, taxing attention and working memory, as well as information overload, when data exceed the human "bandwidth". The ability to achieve high SA in the face of such conditions, for effective decision making and information exploitation, poses a major challenge for interactive systems, requiring new systematic approaches and tools.

In this work, we introduce a novel computational methodology for the real-time dynamic adaptation of UIs, whose objective is to enhance the SA of users by leveraging contextual factors to deliver suitable information, in an optimal and efficient manner. Combining Ontology modeling and reasoning with Combinatorial Optimization, our approach decides *what* information to present, *when* to present it, *where* to visualize it in the display, and *how*, taking into consideration the current context as well as placement constraints (e.g. GUI element overlapping). The main

goal of the proposed approach is to optimize the SA associated with the displayed UI *at run-time*, while avoiding "SA daemons", such as information overload and induced stress. In this respect, a Convolutional Neural Network (CNN) was also developed for the stress detection and 3-level stress classification tasks, achieving state-of-the-art results.

Some of the strengths of our approach, compared to existing ones, include the following: Parameters of the optimization problem are dynamically inferred, based on the current situation through Ontology reasoning. Furthermore, the optimization formulation considers all dimensions of the visualization decision (what, when, how, where), solving layout and GUI element selection decisions simultaneously; this not only provides a more concise handling of the design decisions, but can also lead to improved decisions that deal with the problem as a whole; examples of the benefits of such an approach are better display space utilization and content adjustment based on positional constraints. Finally, the layout of the UI is dynamically defined, with the positions of the graphical elements being dynamically allocated, depending on the current scene.

Our proposed methodology is general-purpose, applicable to different platforms and domains, including desktop, mobile and XR applications, for a variety of potential end-users. In this work, we deploy our computational approach in the context of the European Union funded project *DARLENE* [13], which investigates means by which Augmented Reality (AR) and Machine Learning (ML) can be employed, in real time, to improve the SA when responding to criminal and terrorist activities. Considering the challenges law enforcement and security face today, more efficient ways are required for delivering crucial information meant to aid decision-making in high-pressure and dynamic situations. AR holds massive potential in enhancing the SA of police officers by supplying relevant information, instantly applicable to a given task or situation. Our methodology aims to aid Law Enforcement Agents (LEAs) in making more informed and rapid decisions, through in-situ dynamic adaptivity of the AR display, taking into account the variety of user characteristics, environmental and system factors, as well as the current task.

In order to extract user requirements and model the *DARLENE* application domain, 3 co-creation workshops, with 30 LEAs from police agencies of 5 countries have been organized, gaining insights into context factors that impact the SA of LEAs, and identifying GUI elements that would increase their SA during policing in different tasks and contexts. Based on the analysis of these requirements, an Ontology model has been created, and appropriate inference rules have been defined that take relevant context factors into consideration. Moreover, an optimization problem was formulated, which determines the adaptation of the AR UI. To explore the benefits and limitations of the developed system, two evaluations have been conducted. The first one was an expert-based evaluation with 10 LEAs and User Experience (UX) experts, assessing the appropriateness of the system's decisions, with respect what information was displayed, how detailed it was and where it was positioned. The second one was a user-based evaluation involving 20 LEAs from different agencies, estimating the SA, the mental workload and the overall

UX associated with the system, through an AR simulation. Acknowledging the influence of stress in SA, we assessed our approach and its adaptive capabilities both at normal stress states and under experimentally induced stress.

The results in both states indicate that, in both conditions, the system enhances SA, and while not imposing workload, it provides an overall positive UX. More specifically, cumulatively for both stress states, use of the system improved perceived and observed SA, by 25.63% and 9.25% respectively. In particular, in the case of stressful conditions, perceived SA was improved by 30%, whereas observed SA by 3.95%. In non-stressful conditions, perceived SA was improved by 15.65%, while observed SA by 15%.

This work is organized as follows: In chapter 2, we provide background information in the areas related to our approach and highlight relevant existing literature; in chapter 3, we analyze our methodology; in chapters 4 and 5 we explain the procedure followed and the results in assessing our computational approach in the context of the system's application domain, through an expert-based and user-based evaluation, respectively; finally in chapter 6, we describe the conclusions of our work, as well as future directions.

Chapter 2

Related work

This section carries out a review of related work, elaborating on topics relevant to this thesis, and in particular: context awareness, a key dimension of intelligent and adaptive UIs; context modeling and reasoning, as well as context-aware adaptive UIs, as the main constituents of the proposed approach; combinatorial optimization for UI generation and adaptation, an emerging technique for the automatic generation and adaptation of UIs; and finally, context-aware Mixed Reality, which constitutes the application domain of this work.

2.1 Context and Context-Awareness

The concepts of "context" and "context-awareness" have been present in the literature since the early 90s, and are employed in a continuously expanding set of research areas, including desktop and mobile computing, web computing, ubiquitous computing and IoT, and more recently, IUIs and XR applications.

The terms "context" and "context-awareness" were first introduced by Schilit et al. in [133] where they propose software that examines and reacts to an individual's changing context and classifies it into different categories. They define context as "*information extracted from the environmental entities such as location, people, objects and changes to those objects*" and classify a context-aware system as one that "*can adapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time over the course of the day*". The aspects of context, namely the context factors, they consider important are: where you are, who you are with, and what resources are nearby.

Subsequently, several work followed, proposing different definitions and explanations of context and context- awareness (e.g. [6, 5, 28, 42, 56, 57, 124, 133, 134, 153]). Abowd et al. [5] provide a comprehensive review and develop their own definitions and categorizations to be used prescriptively in the context-aware computing field. According to this work, context is "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application,*

including the user and applications themselves". A system is context-aware "*if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*". This work also introduces the primary context factors of location, identity, activity and time, in order to address the questions of where, who, what, and when.

Context-awareness is particularly significant for Human-Computer Interaction, considering the diverse nature of human bio-psycho-social characteristics, situations of use, and computing platforms. One-size-fits-all approaches cannot handle this inherent context variability leading to bad user experience (UX)[9]. Regarding our proposed system, given the diversity of user's characteristics and conditions and the dynamically varying situations, under which an appropriate UI is generated, context-awareness is inherently required. The context-factors we take into account for the *DARLENE* application domain are user profile and state, and in particular their stress level and expertise, environmental factors, namely if the environment is crowded, system factors, namely the resolution of the display, as well as the current LEA task, for example patrolling.

2.2 Context Modeling and Reasoning for Context Awareness

Representing and reasoning about context is a requisite for understanding and utilizing the different characteristics, properties and details of extracted contextual information. The research community acknowledges that the development of context-aware systems and applications requires appropriate and adequate context modeling and reasoning techniques, which enable high expressivity and maintainability, reduce system complexity and are computationally efficient. In the domain of context-aware systems and UIs, different modeling approaches and inference mechanisms have been proposed in the literature, to represent and reason about context information. The surveys of Strang et al. [144], C. Bettini et al. [23], Perttunen et al. [126], Bikakis et al. [24], Zhang et al. [157], Alegre et al. [12], and Pradeep et al. [127], provide reviews of the most common and prominent approaches to modeling and/or reasoning about context.

2.2.1 Context Modeling approaches

According to Henricksen et al: [80] " *context model identifies a concrete subset of the context that is realistically attainable from sensors, applications and users and able to be exploited in the execution of the task. The context model that is employed by a given context-aware application is usually explicitly specified by the application developer, but may evolve over time*". In general, a model is a representation of abstract concepts, as well as their properties and relationships between them. The modeling process translates real word concepts into modeling constructs [12]. In the case of context modeling, obtained information regarding the current context

is translated to a meaningful for the application value and form, and stored for use. These models require (a) the capability of expressing a variety of complex entities and their relations, (b) to be able to uniquely identify the various contexts and entities, (c) to be simple, reusable, flexible and expandable, and (d) support for uncertainty, incomplete information and validation [23, 126].

In literature, there exists a variety of context modeling techniques for context-aware systems. These can mostly be classified according to the data structure utilized for this purpose. The most prevalent approaches include key-value modeling, markup scheme modeling, graphical modeling, object oriented modeling, logic based modeling and ontology based modeling. A brief introduction of the aforementioned context modeling approaches is provided in the following paragraphs, with an emphasis on ontology based modeling, which is the modeling approach we employ.

Key-Value Modeling Key-value pairs are one of the oldest approaches and are considered the simplest data structure for modeling context. In [133], Schilit et al. proposes employing key-value pairs, which are used as environment variables to the system, in order to define attributes and their values for representing contextual information. In spite of its simplicity and ease-of-use, this technique doesn't support reasoning and lacks the capability of conveniently representing complex structures and relationships, employing only simple value matching checking.

Markup Scheme Modeling Markup scheme models utilize various markup languages, such as the well-known XML, which are derivatives of the Generic Standard Markup Language (SGML) [146]. They allow the definition of markup tags with attributes and content in a hierarchical and usually recursive manner. Representative context modeling approaches of this type are profiles, and in particular extensions of the User Agent Profile (UAProf) [149] standard, which captures information concerning device capabilities and preferences for wireless devices, and the Composite Capabilities / Preferences Profile (CC/PP) standard [1], which describes device capabilities and user preferences and guides the adaptation of content presented to devices. Similar to key-value modeling approaches, mark up scheme modeling exhibits limited capabilities, as shown in [91, 144]. Some of these limitations are with respect to (a) capturing a variety of context types, relationships, dependencies and constraints, (b) inconsistency checking, and (c) reasoning support.

Graphical Modeling Graphical context models express contextual information using graphical diagrams. Representative examples of this modeling approach include models based on the Unified Modeling Language (UML) [55], the Object-Role Model (ORM) [74] and the Entity-Relationship Model (ERM) [34]. UML is a standardized, general-purpose modeling language, appropriate also for modeling context, as demonstrated in the air traffic management application in [21]. ERM

[1] https://www.w3.org/Mobile/CCPP/

and ORM are techniques for modeling and querying databases, but are also extended for modeling contextual information, as shown in [82]. A variant of ORM is the Context Modeling Language (CML), introduced in [83] and refined in [81, 79], which was developed for the conceptual modeling of databases. It extends ORM in capturing different classes and sources of context, imperfect information, dependencies and histories of context. In general, graphical context models facilitate the design and analysis of context, support reasoning, in particular the evaluation of simple assertions and SQL-like queries, and are able to capture incomplete and historical information. However, they exhibit some weaknesses, such as lack of interoperability and support for hierarchical structures.

Object Oriented Modeling In Object Oriented Modeling, the main features of the object oriented paradigm are employed, namely abstraction, encapsulation, inheritance and reusability, through class hierarchies and relationships for context representation. For this type of context modeling, high level object oriented languages are suitable. Representative approaches of this kind are the cues, introduced in the TEA system [135], which provide an abstraction from physical and logical sensors, and the Active Object Model of the GUIDE system [35], which focuses on user location. Usually, the processing of context is encapsulated at the object level, and access to context information is only provided to instances, through inheritance. Object Oriented Modeling benefits from being highly compositional and scalable, maintaining context information in a distributed manner. However, it is considered to have high demands in computational resources and high complexity for development.

Logic Based Modeling In Logic based Modeling, context in represented using facts, rules, logical expressions and variables. Such context models are usually characterized by a high level of formality. In most cases, contextual information is either defined and handled in the form of facts, or derived from specified rules. One of the earliest approaches was introduced in [112, 111], where context information is modeled as abstract mathematical entities, called 'first class objects', with capabilities for fact representation and reasoning useful for AI programs using logic. An example in the domain of Interactive applications is the "Sensed Context Model" proposed in [68], which uses first-order predicate logic to formulate sensed contextual propositions and relations in a formal representation. Although logic based modeling supports reasoning, they exhibit limitations in validation and standardization as well as modeling ambiguity and incompleteness of context.

2.2.2 Ontology Based Modeling

An ontology is a formal description of the concepts and relationships present in a given domain. In [70], Gruber defined an ontology as "*an explicit specification of a conceptualization*" [41]. A more precise and elaborate definition is provided in [145] by Studer et al, based on the definitions of [25, 70]: "*An ontology is a formal,*

explicit specification of a shared conceptualization. A 'conceptualization' refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. 'Formal' refers to the fact that the ontology should be machine readable, which excludes natural language. 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group".

In ontology based modeling, context is modeled with an ontology, and represented through the use of semantic ontology languages and frameworks, such as the W3C Web Ontology Language (OWL)[2], the Resource Description Framework (RDF)[3] , and the Resource Description Framework Schema (RDFS)[4] [3]. OWL is the prevalent one, being more expressive [1] and facilitating greater machine interpretability of Web content in comparison to XML, RDF, and RDFS, through additional vocabulary as well as formal semantics[5]. These languages and frameworks are part of the W3C's Semantic Web technology stack, which realizes the Sematic Web, a new paradigm that defines semantics of information in the web. Ontologies have mainly three components: classes, properties, and individuals [86]. A class represents an entity of a domain, an individual represents an instance of a class and a property establishes a relationship between two individuals or an individual and some value. We will now briefly introduce some notable ontology-based context models that have been proposed in the literature.

The Context Broker Architecture (CoBrA) project [31, 33, 32], introduces a broker-centric agent-based architecture for developing context-aware systems in smart spaces, using an Intelligent Meeting Room as a use case. The OWL language is utilized for the creation of a collection of ontologies to model, share and reason about context information. These ontologies define appropriate semantic concepts and relations for representing physical locations, time, people, software agents, mobile devices and meeting events. The core component of the architecture is the "context broker", which is responsible for (a) storing and sharing a centralized ontology-based context model to the different devices, services and agents of the intelligent space, through its Context Knowledge Base component, (b) reasoning over the stored contextual information, through its Context Reasoning Engine, (c) acquiring contextual information from sources, through the Context Acquisition Module and (d) enforcing policies for the sharing and usage of context information, through the Policy Management Module.

The Context Ontology (CONON) project [152] proposes an OWL context ontology, with the purpose of modeling and reasoning about context in pervasive computing environments. The context modeling approach they follow models a

[2]https://www.w3.org/OWL/
[3]https://www.w3.org/RDF/
[4]https://www.w3.org/TR/rdf-schema/
[5]https://www.w3.org/TR/owl-features/

set of upper-level entities, including location, person, activity and computational entity, and then extends the model by adding specific concepts in different application domains. For that purpose, they distribute their context model into an upper ontology, which represents general features of basic context entities, and a specific ontology, which is a collection of domain-specific ontologies, which define the details of general concepts and their properties in each sub-domain. Moreover, it supports Ontology reasoning, which allows evaluating the consistency of context, and deducing high-level, implicit context from low-level, explicit context.

Similar to [152], the work in [72] also presents an OWL based model to represent, manipulate and access context information in intelligent environments, using an upper, high-level ontology for general context knowledge and low-level, domain-specific ontologies, for specifying the high-level entities. It also introduces a Service-Oriented Context-Aware Middleware (SOCAM) architecture for the building and rapid prototyping of context-aware services in smart environments. Its main components include (a) the Context Providers, which abstract context information from the different context sources, (b) the Context Interpreter, which provides context reasoning services and maintains context consistency, (c) the Context-aware Services, which utilize context and adapt accordingly, and (d) the Service Locating Service, which enables users and applications to locate and access the Context Providers and the Context Interpreter.

The CoDAMoS (Context-Driven Adaptation of Mobile Services) project [128], proposes an adaptable and extensible ontology for creating context-aware computing infrastructures. It is expressed in the OWL language, and it specifies four basic entities: user, environment, platform and service. Its aim is to provide solutions to challenges regarding in Ambient Intelligence Environments, namely (a) application adaptation, (b) automatic code generation and code mobility, and (c) generation of device specific user interfaces.

The work in [154] introduces a general and extensible context-aware computing ontology (CACOnt) for context modeling and providing reasoning capabilities. The context model follows a two-layer hierarchical approach, being divided into generic context ontologies for the general concepts and domain-specific context ontologies, applicable to different sub-domains. The fundamental context models that are specified for capturing the general context information include the User, Space, Environment, Device and Service models. The model supports inference mechanisms for context inconsistency checking and deduction of implicit information, by providing semantic logics which can be combined with rule-based systems. Moreover, this work presents a semantic similarity-based rule matching algorithm, to compensate for the inability of a set of rules to entirely cover the domain of contexts.

Aguilar et al. in [7] propose CAMeOnto (Context Awareness Meta Ontology modeling), an ontology model used by a reflective middleware for context-aware applications, called CARMiCLOC. It is expressed in the OWL language and it is distributed in a two-level hierarchy of ontologies, a general, independent domain one, and a domain-specific one, similar to [152]. In the first level, six contextual

classes are specified, namely user, activity, time, device, services, and location, based on the principles of 5Ws: who, when, what, where, and why.

Although all the context modeling techniques introduced have their benefits and drawbacks, ontology-based modeling is the most widely adopted approach. The research community appears to agree that Ontologies are a promising formalism for modeling context [23, 69, 144, 150]. In the survey of Strang et al. [144], Ontology Based Modeling fulfill all the requirements introduced in the work, contrary to other context modeling approaches. These requirements were: (a) distributed composition, (b) partial validation, (c) richness and quality of information, (d) incompleteness and ambiguity, (e) level of formality, and (f) applicability to existing environments. As indicated in [23], Ontologies exhibit clear benefits with respect to heterogeneity and interoperability, in comparison to other modeling techniques. Another considerable advantage regarding usability aspects is the existence of fairly sophisticated tools, such as 'Protégé' [6], which support and facilitate the design of ontological context models, making it possible even to developers of limited experience with description logics. A further substantial benefit is the support for querying and reasoning to derive new knowledge based on the existing contextual information and to identify potential inconsistencies. In this context, a profusion of query languages and reasoning tools have been developed. Ontological Reasoning will be further studied in a subsequent, dedicated section. Although Ontologies are a powerful context modeling approach, there are still some limitations, such as modeling temporal aspects. Some of the deficiencies could be mitigated by employing hybrid approaches [23].

2.2.3 Context Reasoning approaches

According to Nurmi et al. [119]: "Context reasoning is deducing new and relevant information to the use of application(s) and user(s) from the various sources of context-data". Once the context is modelled, reasoning is required, in order to acquire a better understanding and deduce new knowledge from the available context information". The most popular approaches for context reasoning include Fuzzy logic, Ontology-based, Probabilistic logic, Rule-based, Supervised learning and Unsupervised learning.

Fuzzy logic Fuzzy logic [156] offers approximate reasoning, having the ability to capture concepts of partial truth. In particular, the truth value of a variable may be any real number from 0 to 1, in contrast to Boolean logic, where it can only take the value 0 or 1. Some examples of reasoning engines that incorporate Fuzzy logic are Fuzzy OWL [143], FiRE fuzzy [140] and f-SHIN [141]. It is useful in representing and reasoning about imprecise notions. However, it lacks validation and quality checking of context information and it's usually not used standalone, but in combination with other reasoning techniques, such as ontology-based and

[6] https://protege.stanford.edu/

rule-based (e.g. [40]).

Ontology-based This approach uses description logic, and reasoning about context can be achieved with information modeled in an ontology. It is implemented by Semantic web languages, such as RDF, RDFS, OWL, as well as the Semantic Web Rule Language (SWRL)[7], which represents rule-based first-order logic (FOL) inference rules, expressed in terms of predefined OWL context knowledge. Some reasoning tools available are FaCT[8], Hermit[9], Pellet[10], and RacerPro[11]. Ontology-based reasoning can be combined with ontology-based context modeling (e.g. [85]), supporting complex context representation and reasoning as well as validation and quality checking. However, its capability for handling ambiguity is limited.

Probabilistic logic This approach bases decisions on the calculation of the probability of events and facts. It employs the creation of rules that reason about events' probabilities based on the probabilities of other related events. Through these rules, higher-level probabilistic context can be derived and multi-sensor fusion for improved context quality can be achieved. Prominent techniques of this approach are the Dempster Schafer Theory of Evidence (e.g. [11]), which uses data fusion from sensors to calculate the probability of events, and Hidden Markov Models (HMM) (e.g. [131]), predicting the next state by using the current state.

Rule-based This approach is utilized to expresses policies, constraints, and preferences, regarding context. It usually follows an IF-THEN-ELSE structure, but it can also be based on simple mapping associations of IDs to entities (RFID) (e.g. [105]). Rattanasawad et al provides a review and comparison of rule-based inference engines and languages [129]. Some of the most widely-used ones include the following: FOL-RuleML[12] , SWRL , RIF[13] , Notation3[14] and Jena[15] rules. Rule-based approaches can also be combined with ontological reasoning, such as in the case of the SWRL language. They are easily understandable, widespread and often integrated with ontology models. However, they lack the capability for validation, quality checking, as well as handling ambiguity and incompleteness.

Supervised learning This approach infers new context information, by utilizing a significant amount of labeled training data to build a model that maps new data to output values (regression), or a set of possible outcomes (classification). The

[7]https://www.w3.org/Submission/SWRL/
[8]http://owl.cs.manchester.ac.uk/tools/fact/
[9]http://www.hermit-reasoner.com/
[10]https://www.w3.org/2001/sw/wiki/Pellet
[11]https://www.w3.org/2001/sw/wiki/RacerPro
[12]https://www.w3.org/Submission/FOL-RuleML/
[13]https://www.w3.org/2001/sw/wiki/RIF
[14]https://www.w3.org/TeamSubmission/n3/
[15]https://jena.apache.org/

most prominent techniques are Artificial Neural Network s (ANNs) (e.g. [17]), Bayesian Networks (e.g. [103]), Case-based Reasoning (e.g. [108]), Decision Trees (e.g. [88]) and Support Vector Machines (SVMs) (e.g. [26]).

Unsupervised learning This approach infers new context information without having labeled training data and even without knowing the set of possible outcomes (classes). It tries to extract meaning and find hidden patterns and structures in unlabeled data. Some of the most widespread techniques include Clustering and Kohonen Self Organizing Map (KSOM) (e.g. [151]).

2.3 Context-Aware Adaptive UIs

The aforementioned context modeling and reasoning approaches for context awareness are utilized by a wide spectrum of application domains. In this section, we will focus on approaches supporting the adaptation of User Interfaces, through context awareness.

The user interface is a dominant component of interactive systems and applications, greatly affecting their efficacy, acceptance and the user experience of users. Designing a suitable, user-friendly UI poses many challenges, given the heterogeneity of the possible contexts under which a user interacts with them. This heterogeneity stems from the multiplicity and diversity of the end users, the environmental conditions and the devices. This context variability cannot be sufficiently addressed by constructing various different UIs for the same functionality, since it can lead to an exponentially large number of potential adaptations. At the same time, it cannot be handled by a one-size-fits-all design, but needs to be addressed by adapting the UI so that it is tailored to the current user and the situation surrounding them.

According to Soui et al. [142], "Adaptation is the process of selection, generation or modification of content (e.g., text, image, etc.) to suit the user's profile, its interaction platform and its computing environment. The adaptation of the user interface has been promoted to solve usability problems and to satisfy users' needs and preferences, because it can be performed on the interface container's presentation, such as layout, colors, sizes, and other design elements, and also on the content like data, information, and document." UI adaptation is divided into two categories, namely adaptability and adaptivity. The difference between the two notions, is that, in the case of adaptability, the user participates in the adaptation and tailors the UI to their needs and preferences, whereas in the case of adaptivity, the adaptation is performed automatically, without the need of user intervention and feedback. In this section we will focus on adaptivity, which is the concept relevant to our proposed approach.

Regarding mobile and desktop applications, and web pages, context and context-awareness have been thoroughly investigated, though a profusion of reviews, e.g. [2, 67, 104, 115], reference architectures and frameworks e.g. [14, 16, 29, 30, 37,

41, 53, 59, 109, 115, 125], and adaptation techniques of systems and models that adapt the UIs based on the context of use, taking into consideration context factors related to the user, their environment, their task or the system platform. In the following section, we will introduce a selection of popular model-based frameworks for supporting context-aware UI adaptation. Afterwards, we will introduce UI adaptation techniques and focus on Optimization-based methods.

2.3.1 Adaptive UI frameworks

This section introduces reference architectures or frameworks for adapting UIs of interactive systems. Similar to our approach, they model the relevant context factors, based on which they carry out the UI adaptation. However, they aren't able to support complex adaptation decisions that, for instance, take into account the dynamic nature of the user's field of view, and most of them remain on a conceptual level.

CAMELEON-RT [20] is a conceptual reference architecture for developing distributed, migratable and plastic UI's. It supports adding new adaptive behavior at run-time, through the use of open-adaptive components.

TriPlet [115], is a conceptual framework for context-aware adaptation of UIs. It consists of three core components: a Context-Aware Meta-model (CAMM), a Context-Aware Reference Framework (CARF) and a Context-aware Design Space (CADS). CAMM covers the complete adaptation process, through defining the required concepts, their relationships and properties, necessary for a context-aware application. CARF is a reference framework whose purpose is to define the most relevant concepts for context-aware applications and extensively list and possibilities for implementation and execution. CADS supports stakeholders in the implementation, analysis and evaluation phases of adaptive and adaptable applications, thus covering their complete development life cycle.

CEDAR [10], proposes an approach for developing adaptive model-driven UIS, by introducing the CHEDAR Architecture, the Role-Based UI Simplification (RBUIS) mechanism, and Cedar Studio, which is the supporting ID. The CHEDAR Architecture provides a reference for stakeholders, whereas RBUIS, is role-based mechanism for providing end-users with a minimal feature-set and an optimal layout based on the context-of-use. CEDAR studio supports the development with visual-design and code-editing tools.

AUI-UXA [89] proposes a framework in the form of an adaptive UI/UX authoring tool. It supports UI adaptation based on context factors, such as user disabilities, environment (e.g. light level, noise level, and location) and device, at runtime using adaptation rules for UI rendering. It includes a user model, storing information related to user characteristics and user experience, a context model, storing information about the context, and a device model, storing information about device characteristics. The user can create rules in the form of Conditions-Actions based on the modeled concepts.

2.3.2 Adaptation techniques

Research for adapting UIs is focused on the development of handcrafted rules and heuristics [89]. The creation of rules is carried out either with the help of UX experts, or system designers, or automatically deducted by user interaction data with the system. Furthermore, most of adaptive UI systems use Ontology models, for the purpose of storing the information for tailoring the UI. We will briefly present some examples of such rule-based approaches.

In the method and set of tools presented in [63], end users without programming experience can customize the application UI and/or logic, using trigger-action rules. Within such rules, triggers are associated with dynamic changes for a targeted context of use. The actions are performed when a trigger is verified, and indicate changes in order to achieve the expected adaptation. These rules can be created, saved for use by other users, modified and deleted by the users.

The ISATINE framework [107] proposed a multi-agent adaptation engine, where the adaptation rules are explicitly encoded in a knowledge base, from which they can be retrieved on demand and executed. The application of adaptation rules is ensured by examining the definition of each adaptation rule and interpreting them at run-time, based on a graph transformation system. More specifically, rules are evaluated with respect to a set of metrics. An agent will try to execute them starting from the highest scoring one, while checking that it meets a usability trade-off. This usability trade-off defines relatively the usability criteria that should be preserved when adapting the UI.

The work in [142], presents an ontology-based approach for automatically suggesting adaptive UIs according to the context of use, using SWRL rules. The methodology followed consists of three phases, (a) finding the correlation between the context criteria and the interface characteristics, (b) modeling the adaptation process through an Ontology, which defines the necessary concepts, properties and SWRL adaptation rules and (c) ontological reasoning for the adaptation of the UI, proposing an interface according to the user profile.

Apart from these rule-based or heuristic approaches, Combinatorial Optimization has been proposed as a general purpose method for the automatic generation and adaptation of UIs. In the next section, we will explore this adaptation technique, also adopted by our proposed approach.

2.4 Combinatorial optimization for UI generation and adaptation

An emerging technique for the automatic generation and adaptation of UIs, is Combinatorial Optimization (CO). In [123, 122] Oulasvirta et al. provide comprehensive surveys of combinatorial optimization approaches for graphical user interface (GUI) design, reviewing progress and challenges in UI optimization. They present CO as a powerful yet flexible tool, providing a coherent formalism which

allows identifying problem types, analyzing their complexity, and exploiting known algorithmic solutions. In general, this method treats interface adaptation or generation as an optimization problem, by defining constraints and maximizing (or minimizing) an objective function that represents the goal of the UI, for instance maximizing the interface's usability [106], or minimizing user effort or selection time e.g. [18]. Oulasvirta et al, in [123], categorizes UI design CO problems present in the literature in 7 main problem classes and presents elementary integer programming formulations for them. From these problem classes, the ones relevant to our approach are selection and layouts.

Selection problems emerge in GUI design when selecting which functionality to include. They are concerned with choosing a set of predefined elements which optimize some objective function(s), while at the same time satisfying given requirements. In particular, our optimization problem is a version of a well-known selection problem, the knapsack problem [110]. In the knapsack problem, we are given a set of elements U = u1, . . . , un, each with weight wi and value vi,, along with a maximum weight capacity W. The goal is to find a selection of the elements that does not exceed W and maximizes the total value. More specifically, we are interested in the 0-1 knapsack problem, where the copies of each kind of item are zero or one.

Layout problems involve fitting a set of given objects onto a canvas, while satisfying feasibility constraints, such that there is no overlap or overflow. This problem is usually solved for 2D geometric layouts, where a set of rectangles or bounding boxes need to be placed within a 2D canvas. For the formulation of layout problems, a disjunction of constraints is typically defined which expresses that, in order to avoid an overlap of two rectangles one of them has to be to the left of or above the other. However, we follow a different formulation approach since, in our case, it is not sufficient that the rectangles don't overlap, but each one of them can occupy specific positions, potentially relative to others.

Our optimization problem combines characteristics of both selection and layout problems. Similar to the knapsack problem, it tries to select the content and presentation that maximize the total value, which in our application domain is the total SA associated with the UI, given the context. At the same time, this selection is also constrained, in our case by the maximum number of displayed graphical elements, depending on the current context, to avoid information overload and induced stress. In parallel to the selection problem, our optimization problem tries to solve a layout problem, since it also aims to determine, in which of the available positions to place the graphical elements, without overlapping with others. These selection and layout dimensions are being solved simultaneously, the one affecting the outcome of the other.

Regarding the applications of CO for UI design, Oulasvirta et al, in [123], divides them in 4 types, generative design, interactive design, adaptation and personalization, and decision support. In generative design, the objective is to algorithmically generate high quality UIs that are usable, reliable and can recover

from errors [117]. Interactive design encompasses human-in-the-loop or optimizer-in-the-loop methods that aid designers in different ways, such as exploring design decisions, comparing UIs, as well as completing partial designs. In decision support applications, CO informs UI decision making by, for example, applying bounds to solution quality and assessing robustness in changes in assumptions. Adaptation and Personalization applications use CO to adapt UIs, whose associated objective function captures individual-specific or dynamic requirements.

Our approach belongs to the "Adaptation and Personalization" category, adapting the UI based on the user and the current context of use. A first approach towards this direction was SUPPLE [58], which revolutionized the field of adaptive UIs, by proposing UI adaptation as an optimization problem. It utilizes input traces of typical user behavior, to adapt the UI to the specific user. The work presented in [60] focuses on ability-based optimization, where UIs are adapted by considering the user's motor or cognitive impairments. SUPPLE++ [60] is a system which can automatically generate UIs adapted for motor and vision-impaired users. It uses custom models of motor performance (Fitts' law) and heuristic models of human vision (rules-of-thumb), for use in the optimization process, generating a personalized interface. Sarcar et al. [132] explores a computational design approach using CO for improving UI designs for users with sensorimotor and cognitive impairments. To this end, this work presents the Touch-WLM predictive model, that predicts text entry on touchscreen devices by users with given disabilities, and describes how impairments, and in particular tremor, dyslexia, and memory deficits, can be incorporated in the model.

Existing approaches are mainly focused on design-time adaptation of the UI at startup, as opposed to real-time continuous adaptation based on the current context. However, UIs are nowadays increasingly being used in constantly changing contexts, such as in mobile and MR applications, calling for more dynamic approaches. Lindlbauer et al. [106] proposed an optimization-based approach for adapting Mixed Reality UIs at run-time, based on the current context, and in particular the users' current cognitive load and task. More specifically, it adapts which applications are displayed, how much information they show, and where they are placed in the UI. Similar to their work, our approach uses CO to dynamically adapt Mixed Reality UIs, in line with the current user profile, state, task and environment, adjusting which information is displayed, at which detail, and where in the UI. To this end, it incorporates a novel combination of Ontology modeling and CO, where parameters of the optimization problem are reasoned at run-time from the context model, contrary to the manual, hard-coded approach of the state-of-the-art.

2.5 Context-Aware Mixed Reality

With respect to Mixed Reality systems, there is an increasing number of application domains for context-aware Mixed Reality applications. In the domain of

maintenance assistance, there exist various systems that apply AR to facilitate routine and ad hoc maintenance activities, e.g. [158]. Moreover, there is a growing number of systems and applications in the field of medicine, e.g. [15, 44, 94], to assist in surgical operations and training, as well as in the fields of cultural heritage, e.g. [38, 62, 97], and education, e.g. [95, 96]. To the best of our knowledge this is the first context-aware AR system for Law Enforcement, aiming to aid the decision-making of LEAs by enhancing their SA.

In general, there exist different frameworks, e.g. [92, 100, 155], and applications, e.g. [64, 106], targeting the area of context-aware Mixed Reality. Oh et al. presented CAMAR [120] (Context-aware Mobile Augmented Reality), a conceptual approach to adapt the virtual content in mobile AR, based on the user's profile, and identified key technical challenges towards this approach. Zhu et al. [158] proposed an authorable context-aware AR system (ACARS), to assist the maintenance technicians. It aims to adapt the content, format and location of maintenance instructions presented in the UI, based on the context of use, considering factors such as the user, their location, their activity, their device and their equipment. To this end, an ontology-based context model is constructed, on top of which appropriate rendering rules are defined using SWRL rules. Grubert et al. introduced the concept of Pervasive Augmented Reality (PAR) [71], emphasizing the importance of sensing the user's current context and adapting the AR system based on the changing requirements and constraints, for a continuous AR experience. It summarizes existing approaches and presents a taxonomy of context-aware AR, which classifies context sources (e.g. anatomic and physiological state, location, system state) and context targets (e.g. content, appearance, modality) relevant for implementing PAR. Krings et al. presented AARCon [100], a framework for developing context-aware AR applications. It consists of three main components: (a) the Context Monitoring component, responsible for constantly collecting context information, measured by sensors (b) the Adaptation component, responsible for the execution of actions in response to a captured context change, and (c) the Decision Making component, which coordinates the other components, by keeping track of the active conditions based on the current context and corresponding adaptation actions for execution.

Our approach is concerned with dynamically determining what virtual content is displayed, how, and where in the Mixed Reality display. Existing literature mostly addresses what information to display and how, using heuristic or rule-based approaches, where a particular context instance is mapped to a content or presentation style adaptation. For instance, Zhu et al. [158] uses SWRL rules for adapting the virtual content (e.g. item, instruction sub-step) based on factors such as the current task, the user's expertise and device characteristics, as well as adapting its format (e.g. color, transparency) based on characteristics such as the user preference and distance. Ghouaiel et al. proposed adapting the displayed augmentation based on the scene illumination, the distance to the target and the ambient noise, using appropriate formulas [64]. Contrary to previous approaches, Lindlbauer et al. [106] formulates a CO problem whose solution adapts the virtual

content and its presentation based on the users' cognitive load and task. Similar to [106], we use an optimization based approach, but combined with Ontology–based reasoning, which adapts the content of the augmentations and their presentation based on the user's profile, state, task and environment. Focusing on the presentation of information (how), DiVerdi et al. proposed the concept of employing different granularity levels of content, namely LoDs, as a basis for adapting AR UIs [43]. Our system, as well as the works in [106, 158] adopt this concept, using different LoD presentations of content based on the current context.

With respect to the question of automatically determining where to place virtual elements, there is several work addressing it using computational methods. These approaches are related to the problem of view management, as described in Bell et al. [22]. An example for spatial AR is OptiSpace [52], a system for automatically positioning virtual projections, based on measuring users' behavior and the geometry of physical surfaces over time. It aims to optimize content placement, according to perceptual criteria, such as visibility and quality of virtual content. FLARE [61] is an application development system, using a rule-based approach for generating object layouts for AR applications. To this end, it solves a constraint-satisfaction problem in real-time, using extracted features from the physical environment. SnapToReality [118] proposes automatic alignment techniques, which extract real-world constraints, such as linear edges and planar surfaces, from the environment. Its objective is the precise alignment of virtual objects to the real world.

As discussed above, in general, current approaches have focused on automatically placing virtual content based on surface geometry or visibility, without taking the user's context into account. However, the work by Lindlbauer et al. [106] considers the current task, and places virtual elements of greater utility for a specific task at better positions. More specifically, there is a pre-defined set of slots in the field of view that can host elements and for each of the slots, there is a score assigned, corresponding to the position quality. The assignment of virtual elements to slots is performed in a greedy manner, and higher utility elements are assigned to slots with higher quality. In our proposed work, the context of use also affects where graphical elements are placed, by influencing their LoD and size, and thereby their possible positions. However, instead of scored, independent and predefined slots containing the virtual elements, we follow a more-fine grained approach, where the elements occupy non-overlapping grid tiles in the display. Considering the dynamic nature of scene that we are augmenting with virtual content, our approach allows satisfying in real-time positional relationships between graphical elements and placement constraints (e.g. the carried weapon information is displayed relative to the moving detected Foe).

Furthermore, current approaches separate the decision of what graphical elements to place and how, with the problem of where to place them. In the work by Lindlbauer et al. [106], it is first determined by the optimization step which UI elements are displayed and at which LoD and then, as a final step, their placement

is specified. In contrast, we follow a novel approach, where the placement of graphical elements is incorporated in the optimization step. As a result, the decisions of what content to display, how and where, are not made independently, utilizing better the available display space and adjusting the presented content based on positional constraints.

Chapter 3

Methodology

In the context of this work, a general purpose methodology was adopted, for the dynamic adaptation of context-aware UIs, aiming at enhancing the Situational Awareness (SA) of the user. The Decision Maker (DM), developed in this thesis, is the central module of our optimization-based approach and has been deployed in the context of the *DARLENE* system, as the fundamental decision making component for improving the SA of LEAs. It decides, based on context factors, which information will be displayed, how it will be presented, and in which position in the UI. To this end, it interfaces with appropriate input and output modules, in order to acquire the necessary information and to provide its decisions for visualization. These modules include the following:

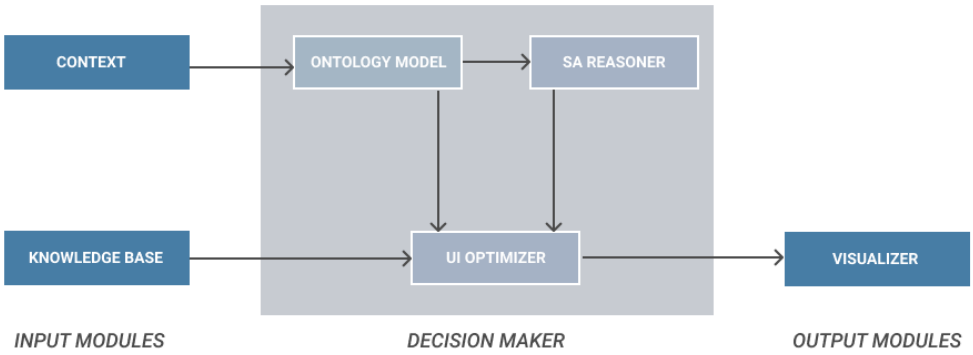

Figure 3.1: The interface modules and the units of the Decision Maker (DM)

- The Context module, which extracts the relevant context information regarding the user's profile (e.g. expertise) and state (e.g. stress level), the environment (e.g. crowdedness) and the task/activity at run-time (e.g. incident resolution), and propagates it to the DM. In the context of this thesis, a Deep Neural Network (DNN) was constructed to detect user stress and, relevant to the *DARLENE* application domain, as presented in section 3.5.

- The Knowledge Base module (KB), which provides to the DM the necessary intelligence (Information Elements) regarding the current scene (e.g. detected humans, identity information) through ML algorithms (e.g. object detection algorithm) and human input (e.g. feedback from the Command & Control). In the context of this thesis, the Knowledge Base module constitutes an external module (it is considered as a black box), which provides DM with the appropriate Information Elements.

- The Visualizer module, which contains the collection of design 'templates' for all the supported GUI elements and performs the rendering in the display. For that purpose, it receives from the DM the rendering configuration of the GUI elements, and in particular, which information to display, with which design 'template' and at which position. Its implementation depends on the application platform.

The DM consists of three inter-connected units:

- The **Ontology Model**, which models through an Ontology (a) the supported GUI elements, along with accompanying metadata (e.g. their dimensions) and (b) the relevant context information. It dynamically receives the current context from the Context module and stores it in the Ontology.

- The **SA Reasoner**, which dynamically quantifies how suitable each GUI element is for display (its SA score), in terms of enhancing the SA of the user; this is based on information from the Ontology Model and, in particular, the current context and modeled domain knowledge in the form of Ontology rules.

- The **UI Optimizer**, which computes the optimal adaptation of the UI, given our modeling of the problem. In particular, it determines the GUI elements, their presentation and their position, for display by the Visualizer module. This is based on information about (a) their SA score provided by the SA Reasoner, and (b) visualization and placement constraints, based on the current context (provided by the Ontology model) and their size and shape.

Detailed descriptions of the aforementioned units are provided in sections 3.2, 3.3 and 3.4 respectively.

The interface modules and the units of the DM are portrayed in Figure 3.1. In short, the flow of information is the following: When the current context changes, the Context module propagates it to the DM, which updates the Ontology Model accordingly. Based on this new state of the Ontology Model, and its intrinsic modeling, the SA Reasoner recalculates the SA scores, and sends them to the UI Optimizer. In parallel, the Knowledge Base sends real-time data in the form of Information Elements to the DM. Each Information Element is translated to potential designs based on the GUI elements modeled in the Ontology and is stored

in the UI Optimizer. At a frequency equal to the Visualizer's rendering frame rate, the UI Optimizer decides which Information Element will be visualized, through which GUI element and where it should be placed, based on the information from the SA Reasoner and the Ontology model. Given this decision, the appropriate rendering configurations are generated and propagated to the Visualizer for display.

In the following sections, our methodology is detailed, consisting of a requirements analysis and elicitation procedure for the target application domain, described in section 3.1, the development of the DM units in sections 3.2, 3.3 and 3.4, and the construction of a stress detection model, in section 3.5.

3.1 Requirements Elicitation and Analysis

The first step of our methodology is to solicit and analyze the user requirements for our target application domain. The goal is to gain insights into the context factors that impact user SA and identify the types of information that would increase it during different situations and tasks. These findings are then utilized by the DM, shaping its functionality and behaviour. In particular, the Ontology is accordingly populated, and appropriate Ontology rules and Optimization constraints are defined, in line with the user requirements.

In order to employ the aforementioned approach in the context of the *DARLENE* project, the requirements elicitation and analysis was carried out through the organization of three Co-creation workshops, which were performed online, since physical presence meetings were not feasible due to the Covid-19 pandemic. In these workshops, a total of 30 end users (LEAs) participated, from police agencies in 5 countries. In the subsequent sections, we describe the procedure followed, as well as our findings.

3.1.1 Workshop procedure

Co-creation workshops are a participatory design method. Co-creation is "*a creative process that taps into the collective potential of groups to generate insights and innovation. Specifically, it is a process, in which teams of diverse stakeholders are actively engaged in a mutually empowering act of collective creativity with experiential and practical outcomes*" [130]. All workshops we conducted had the same structure of activities. First, an introduction was made to the participants, regarding the aims and objectives of the workshop and a warm-up activity took place, whose purpose was to "break the ice" and stimulate discussion within the group. Then, after a presentation of the project and its use cases, to familiarize them with the system concept, the co-creation activities took place. Specifically, they consisted of three phases. At first, the participants identified the functionality they would like the system to have. Afterwards, they voted on the most appealing functionality and at the end an analysis of the top-voted functionality took place. The workshops concluded with an evaluation of the workshop procedure and experience.

3.1.2 Requirements analysis

The workshop's outcomes were analyzed following a combination of deductive and inductive coding, involving two researchers. One code for each functionality identified during the first activity of the workshop was assigned. Then, the researchers examined the entire dataset (from all the workshop activities) in order to identify system requirements and assign to each one a predefined code. In the cases where the need for assigning a new code was identified, this was added to the set of codes, and all responses were re-examined. The examination of responses and the code assignment was carried out by two researchers, individually. The outcomes of the two independent analyses were compared, following a consensus-building approach to address inconsistencies in the codes assigned.

3.1.3 Results

The systematic analysis of the outcomes of the workshops constituted the foundation for specifying the system requirements. In particular, 44 requirements were identified, a subset of which is considered in this first version of the system. This requirements' subset identified types of information (Information Types) that address some of the reported LEA's needs, and context factors that the system should take into account when supplying them. These Information Types, were translated to homonymous Component Types, which correspond to collections of design 'templates' (Components), providing alternative presentations for an Information Type. These Component Types and Components are modeled in our Ontology in section 3.2. The currently supported Component Types and the description are presented in Table 3.1. In addition, context factors currently considered are presented in Table 3.2. We should note that, in addition to the factors that emerged from this analysis, we also consider the device the user is currently utilizing, for the appropriate placement and visualization of the GUI elements.

Table 3.1: List of Component Types and their description

Component Type	**Description**
Foe Detection	Highlights foes (e.g. perpetrators, accomplices) in the LEA's field of view
Suspect Detection	Highlights suspects (e.g. persons behaving oddly) in the LEA's field of view
Criminal Activities	Gives information regarding the criminal activities of a foe or suspect (e.g. killings, kidnappings, theft)
Carried Weapon	Gives information regarding the identity and type of a carried weapon (e.g. gun of type X, knife of type Y, home-made explosives)

Continued on next page

Table 3.1 – *Continued from previous page*

Component Type	**Description**
Unattended Object Detection	Highlights unattended dangerous or potentially dangerous objects (e.g. suitcases, bags) in the LEA's field of view
Object Identity	Gives information regarding the identity of a dangerous or potentially dangerous unattended object in the scene (e.g. unattended bag, suitcase with explosives)
Abnormal Behavior	Gives information regarding the abnormal behavior of a person (e.g. irregular clothing, odd movement)
Ally Detection	Highlights allies (e.g. officers, medical personnel), in the LEA's field of view
Physio Signals	Gives information regarding the physiological signals of an ally (e.g. heart rate, stress)
Victim Detection	Highlights victims (e.g. injured civilians) in the LEA's field of view
Health Status	Gives information regarding the status of a victim (e.g. if they are bleeding)
Health Record	Gives information regarding the health record of a victim (e.g. Diabetes, HTS)
Summative Scene Info	Gives information regarding the number of foes, suspects, allies, weapons and unattended objects in the scene
Map	Represents diagrammatically an area or building and indicates points of interest (e.g. entry and exit points, gas leak, suspects)
Alerts	Gives urgent information from the command and control (e.g. arrest John Doe)
Procedural Information	Gives step-by-step guidance on how to perform a task relevant to the current situation (e.g. stop blood loss, open locker)

Table 3.2: List of Context Factors and their description

Context Factor	**Description**
Stress	The current stress level of the LEA
Crowdedness	The crowdedness level of the environment the LEA is currently located
Expertise	The expertise of the LEA in the field

Continued on next page

Table 3.2 – *Continued from previous page*

Context Factor	Description
Task	The task the LEA is currently executing
Device	The device the LEA is currently utilizing

3.2 Ontology Model

In the Ontology Model unit of the DM, we model the studied application domain, based on the user requirements. To this end, an Ontology is created which captures all the relevant context factors that the system considers, as well as all the supported GUI elements for display, along with accompanying metadata (e.g. their dimensions)

Regarding the *DARLENE* use case, the resulting Ontology captures context information related to the user, and more specifically their profile, their current psychological state, their environment, the task they are executing and their device (HUD). At the same time, it represents in a hierarchical manner the pre-specified GUI elements currently supported for display in the LEA's field of view. Relevant properties of the relationships between the modeled entities are also included in the model. In Figure 3.2 a graphical representation of the Ontology is depicted. The Ontology is logically separated into two parts, the context factors, and the GUI elements, explained in the subsequent subsections 3.2.1 and 3.2.2, respectively.

3.2.1 Context factors

The first part of the Ontology models information about the current context, capturing primary context factors of context-aware systems, namely (a) the user, (b) the activity, (c) the environment, and (d) the device. Concretely, it models with the appropriate entities (a) the profile of the user - LEA, which captures their expertise in the field, as well as their current psychological state, which pertains to their current stress level, (b) the current LEA operation task (i.e., Patrol and Incident Resolution), as defined in the context of the *DARLENE* project, (c) the environment in which it is being executed, and in particular its crowdedness level, and lastly (d) information about the device the LEAs are using, which is the HUD and in particular their AR glasses. More specifically, the necessary information regarding the HUD is modeled in the data properties "Width Resolution", "Height Resolution" and "Sampling Rate". These parameters are used by the UI Optimizer in order to assign appropriate, non-overlapping positions to the displayed GUI elements. In particular, the "Width Resolution" and "Height Resolution" properties store the pixel resolution of the device display, while the "Sampling Rate" stores the rate in pixels by which the display space is sampled for performance reasons, as described in section 3.4. For example, a Sampling Rate of 60px, denotes that we consider every 60th pixel of the display.

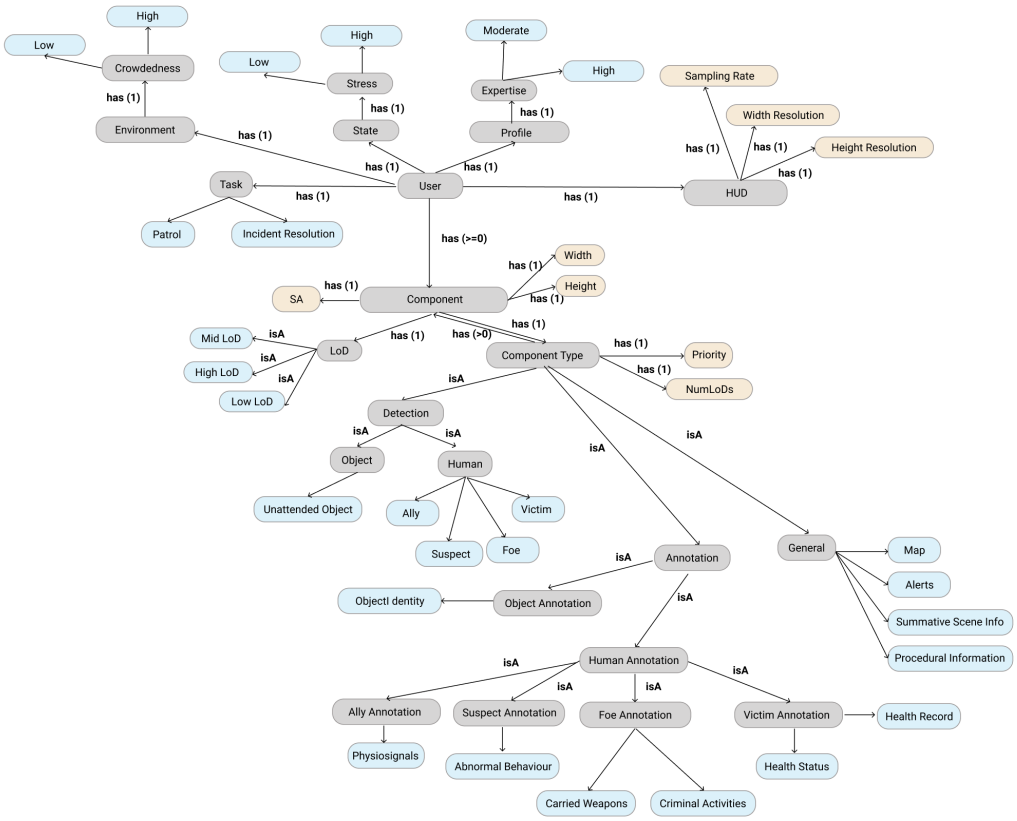

Figure 3.2: System Ontology model

3.2.2 GUI elements

The second logical part of the Ontology represents information about the supported GUI elements. With the term 'GUI elements', we refer to all the graphical entities of our modeling approach, which are conceptually organized in three levels, with increasing level of specificity: (1) Component Types, (2) Components and (3) Component Instances. Each Component Type, listed in Table 3.1, corresponds to a collection of the design 'templates' for a specific Information Type. In particular, it provides alternative presentations (levels of detail) for Information Elements of the corresponding Information Type. Each individual design 'template' belonging to a specific Component Type is a Component. Once a Component is instantiated with an appropriate Information Element from the KB and its position in the display is determined, it becomes a Component Instance. The Component Instances do not need to be included in the Ontology, as explained later in this paragraph. In figure 3.3, we can see the related information and properties for the notions of Component Type, Component and Component Instance, through an example.

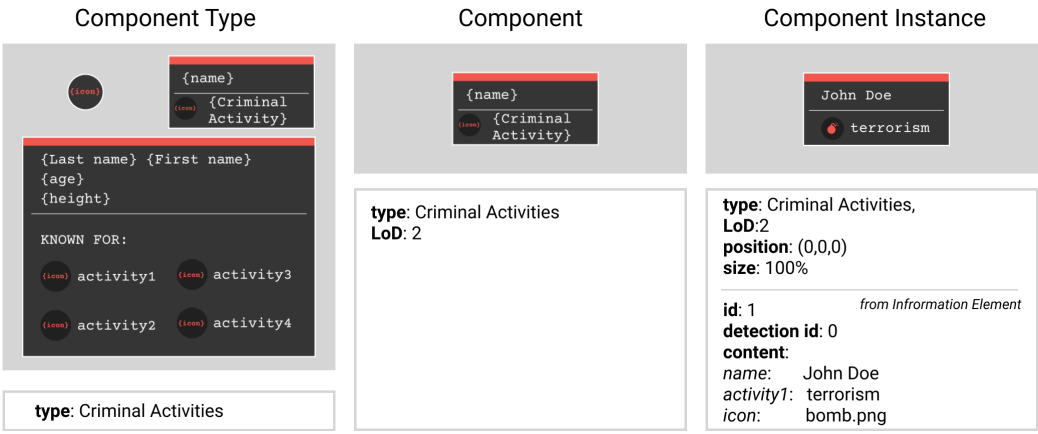

Figure 3.3: An example of the properties corresponding to a Component Type, Component and Component Instance, as well as their relation

Component Types The Component Types are modeled in a hierarchical manner, by first naturally dividing them into three main disjoint categories. The first category is the "Detection" category, which corresponds to an object or person of interest detected in the user's field of view. The purpose of the Component Types of this category is to draw the attention of the user to the detected entities, for instance to an unattended object ("Unattended Object" Component Type), or foe ("Foe" Component Type) by highlighting them appropriately. The second category is the "Annotation" category, whose Component Types depend on a particular detection and provide information for it. For instance, for a detected armed man in the user's view, Component Types of this category can provide information regarding the carried weapons ("Carried Weapons" Component Type) and known

criminal activities (“Criminal Activities" Component Type) of the person. The third and last main category is the “General” category, which consists of Component Types that don’t relate to a specific detection in the scene, contrary to the other two categories, but provide general information regarding the surrounding environment, and events that are taking place. Examples of this category are alerts from the command and control (“Alerts” Component Type), and procedural information (“Procedural Information” Component Type), for instance, on how to stop blood loss on an injured victim. The “Annotation” and “Detection” categories are also divided into two subcategories, “Object Annotation” and “Human Annotation” for the former, and “Object Detection” and “Human Detection” for the latter, based on whether they correspond to a Human or an Object.

Regarding its modeled data properties, a Component Type has a data property “hasPriority”, which is used in the SA Reasoner unit of the DM. It reflects the priority of the Component Type for being displayed, relative to the rest, given the task the user is currently executing. Its purpose and possible values will be analyzed in section 3.3.

Figure 3.4: An example of the 3 LoDs of the Component Type "Criminal Activities", from low to high, instantiated with context, as displayed by the Visualizer

Furthermore, to cater for the different needs, with respect to information quantity and presentation, depending on the current context of the user, each design "template" of a Component Type corresponds to a different granularity level, called “Level of Detail” (LoD). The higher the LoD, the more analytically information is presented. The number of LoDs a Component Type supports are represented in the data property “hasNumberOfLoDs”, and in this version of the system this number ranges from 1 to 3. In Figure 3.4 an example of the 3 LoDs of the Component Type "Criminal Activities" is depicted, instantiated with content (Component Instances), as diplayed by the Visualizer .

It should be noted that, when the KB sends an Information Element to the DM, it gets mapped to the corresponding Component Type, based on its homonymous Information Type. Thus, for the modeling purposes of the Ontology, the entity

'Component Type', is sufficient to represent the input from the KB.

Components We model the GUI elements that have a particular Component Type and LoD as Components. Each one represents a design template, that can host content (e.g. icon, text), when instantiated. Its appearance, namely its specific shape, size, colors and so on, is determined both from their Component Type and their LoD. Thus, the supported Components across Component Types are all the possible design 'templates' that can be displayed as augmentations when initialized. Moreover, each Component is fundamentally a Component Type in a specific LoD, "LowLoD", "MidLoD" or "HighLoD". The data properties of a Component modeled in the Ontology are its width and height, utilized by the UI Optimizer for placement considerations. In most cases, for Components of type "Annotation" and "General", the higher the LoD, the greater the size they occupy in the display. We should note that for Components of type "Detection", we don't have that information, in order to include it in the model at start-up, since the width and height of the bounding box of a detection depends on the actual shape of the human or object. Besides information regarding their size, another modeled data property of Components, perhaps the most significant, is the "hasSA". This field stores a score, the "SA" score, which is computed in the SA Reasoner, based on the Ontology model, as we will see in detail in section 3.3. This SA score models the Component's appropriateness for display in the UI, and, in particular, how much it contributes to an increased SA, given the current context information modeled in the Ontology. Its purpose is to provide the Optimizer with information on which Components to "favor" for display, as described in section 3.4.

Component Instances As already indicated, Components are visualized instantiated in the UI, with specific content (e.g. image, icon, text) that corresponds to the available information (Information Elements) from the KB, and position in the display. This information can be linked to the unfolding events during the LEA's operation, or a specific person or object of interest, in their field of view. The granularity and type of information of these Component Instances depend on the LoD and Component Type of the Component they are instantiating. The higher the LoD, the richer and more descriptive the content is, as we can observe in Figure 3.4. In the case of "Annotation" Component Instances, the content depends on the provided information regarding the detected person or object they refer to, in the case of "General" Component Instances, the content depends on the current situation and events that are taking place, and in the case of "Detection" Component Instances, it depends on the detected humans and objects and, in particular, recognized key-points e.g. joints. Although, eventually, Component Instances are displayed in the UI, in the current version of the system, the decisions are taken at the level of Components and not Component Instances. This means that their content isn't taken into account by the DM for determining the ones to be visualized, but only their Component Type and LoD. As a result, Component Instances don't

need to be modeled in our Ontology, and information regarding their content is directly propagated from the KB to the UI Optimizer, and finally to the Visualizer for rendering.

3.2.3 Ontology definition

In this sub-section, the procedure followed and the tools utilized for the purpose of creating our Ontology are detailed, built in the OWL 2.0 Web Ontology Language. For the purpose of designing and visualizing our Ontology, the software 'Protégé'[1] was utilized, an open-source ontology editor and framework for building intelligent systems. In order to manage the defined Ontology, and perform reasoning based on it in section 3.3, Owlready2[2], a package for ontology-oriented programming in Python was utilized.

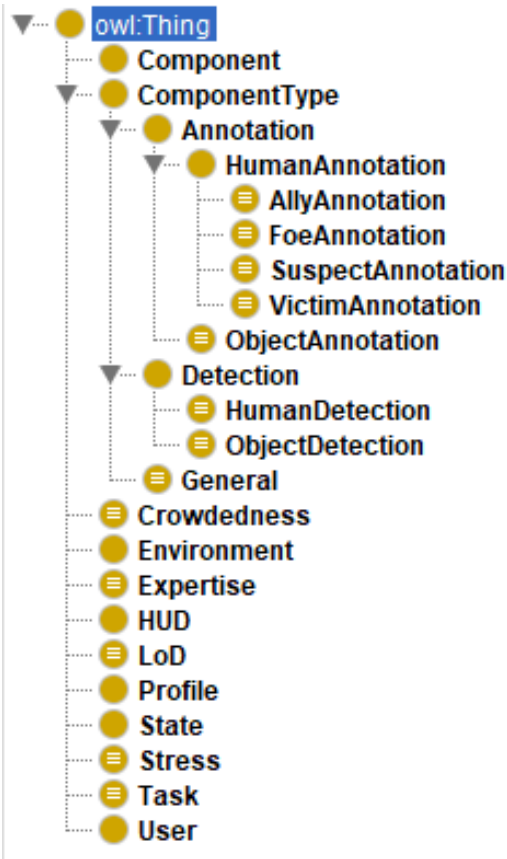

Figure 3.5: Class hierarchy in Protégé

The construction of the Ontology was carried out in the following steps. First, the appropriate Classes were defined in a hierarchical manner, capturing the main

[1]https://protege.stanford.edu/
[2]https://pypi.org/project/Owlready2/

concepts of our application domain. In figure 3.5, we can see the class hierarchy displayed in Protégé. By default, OWL makes an open world assumption, meaning that everything that is not stated in the Ontology is not 'false' but 'possible'. As a result, things and facts that are 'false' or 'impossible' must be clearly stated as so in the Ontology. Therefore, all Classes at the same hierarchy level need to be defined as 'Disjoint', meaning that for each pair of Classes, that don't have an "isA" subclass relationship, an instance cannot belong to both Classes. For example, a Component Type cannot be both an "Annotation" and "Detection", or at the same time an "AllyAnnotation" and a "FoeAnnotation". However, an "AllyAnnotation" is a "HumanAnnotation", which in turn is an "Annotation".

Table 3.3: Object Properties

Domains	**Object Properties**	**Ranges**
User	hasState	State
State	hasStress	Stress
User	hasProfile	Profile
Profile	hasExpertise	Expertise
User	hasHUD	HUD
User	hasTask	Task
User	hasEnvironment	Environment
Environment	hasCrowdedness	Crowdedness
User	hasComponent	Component
Component	hasLoD	LoD
Component	hasComponentType	ComponentType

Then, the appropriate Object Properties were specified, which express the relationships between the Classes. In Table 3.3 the list of Object Properties is summarized, including their Domains (represents the "subject", answering to the question who?) and Ranges (represents the "object", answering to the question what?). Regarding the characteristics of the Object Properties, all except the "hasComponent" are functional, meaning that they connect a single instance of the Range Class for a given instance of the Domain Class. For example, a "User" is carrying out one "Task" at a time, at one specific "Environment", wearing one "HUD", so the Object Properties "hasTask", "hasEnvironment", and "hasHUD" are functional, but can have many components, so the Object Property "hasComponent" isn't functional. All Object Properties, are Disjoint, meaning they are different properties, expressing different relationships.

Table 3.4: Data Properties

Domains	**Object Properties**	**Ranges**
HUD	hasTileDimention	int
HUD	hasDisplayWidth	int
HUD	hasDisplayHeight	int
Component	hasSA	float
Component	hasWidth	int
Component	hasHeight	int
ComponentType	hasPriority	int
ComponentType	hasNumberOfLoDs	int

Afterwards, the appropriate Data Properties were defined, which similar to Object Properties express relationships, but with a Data Type (integer, float, etc.), instead of another Class in their Range. In Table 3.4 we can see the list of Data Properties, with their Domains and Ranges. With respect to the characteristics of the Data Properties, all are functional, taking a single Data Type value for a given instance of the Domain Class.

Table 3.5: Possible values of Classes

Class	**Possible Values**
Stress	HighStress, LowStress
Expertise	ModerateExpertise, HighExpertise
Crowdedness	LowCrowdedness, HighCrowdedness
Task	Patrol, IncidentResolution
LoD	LowLoD, MidLoD, HighLoD
ObjectDetection	UnattendedObject
HumanDetection	Foe, Suspect¸ Ally, Victim
ObjectAnnotation	ObjectIdentity
AllyAnnotation	Physiosignals
SuspectAnnotation	AbnormalBehaviour
FoeAnnotation	CarriedWeapons, CriminalActivities
VictimAnnotation	HealthStatus, HealthRecord
General	SummativeSceneInfo, Map, Alerts, ProceduralInfo

Finally, the possible values of Classes were enumerated as Instances of the Class. This is a way to define enumerations in an Ontology, and in particular, by specifying that a class is equivalent to one of a list of Instances. In Table 3.5 we

can see the list of possible values supported for different Classes, in the current version of the system. Given the open world assumption, all Instances are declared as different from each other.

After the Ontology was constructed, it was populated by the supported Components, as Ontology Instances of the Component Class. This was carried out by creating an Instance for every valid LoD for each Component Type.

3.3 SA reasoner

In the SA Reasoner unit of the DM, an SA score for each Component in the Ontology of 3.2 is dynamically computed, depending on the current context. More specifically, based on the user's profile, state, activity and environment, modeled in the Ontology, an Ontology Reasoner infers the SA score to assign to each Component, depending on its Component Type and LoD. Specifically, for the *DARLENE* case study, the activity is the current LEA operation (task), and the user state we are interested in is with regard to Stress. Each time the context changes, the SA Reasoner recalculates the SA scores and propagates them to the UI Optimizer, described in section 3.4.

SA Score concept This SA score represents how suitable a Component is to be displayed in the UI, and in particular, how much it contributes to an increased SA of the user, in comparison to the other Components, given the current contextual information. The purpose of the score is not to express some measured or formally computed SA value, but to provide a weak ordering of the Components so that the ones that are more appropriate and abler to enhance the SA of the user acquire higher score relative to the rest. More specifically, its goal is to provide the UI Optimizer with the necessary information as coefficients for each Component, so that the latter will be able to decide which Component Instances to display, so that the usefulness of the UI in terms of the associated SA is maximized.

The SA score of a Component, given the context, depends on its LoD and Component Type. To provide an example of this dependency with the LoD, in Figure 3.4, the Component with LoD 'Low' is only an icon representing the criminal activity a given foe is mostly known for, whereas in LoD 'High', it also provides identity information (e.g. name) and a more detailed criminal record. Although the higher LoD may empower the user with more information, enhancing their SA under favorable physiological and environmental conditions, in other situations, such as of high stress and crowdedness level, textual and more detailed information that obscures more space in the user's field of view may potentially have a negative impact on their SA, and lead to information overload and induced stress. Thus, in the former cases of context, the SA score of the Component is higher for LoD 'High', whereas in the latter, it is higher for LoD 'Low'. Moreover, depending on the current task, some Component Types may be more appropriate and useful to display than others. For instance, in our LEA application domain, providing

information about the carried weapons of a foe ("Carried Weapons" Component Type) during an "Incident Resolution" task can lead to higher SA for the LEA, than providing procedural information ("Procedural Information" Component Type). As a result, the former Component Type has higher SA score than the latter, for this task.

SWRL Rules definition For our *DARLENE* use case, in order to assign SA scores to Components, depending on the context, feedback from end-users has been utilized, obtained through the virtual Co-Creation Workshops and a subsequent questionnaire. This feedback led to the specification of a set of SWRL rules, through which an Ontology Reasoner reasons about the appropriate SA score, given the context. In our system implementation, the Reasoner 'Pellet'[3] OWL 2 reasoner was employed for this purpose. These rules can be divided into two categories. The first category, the 'Priority SWRL Rules', assigns a 'Priority value' to the Component Types, based on the task the LEA is currently executing. The second category, the 'SA SWRL Rules', assigns the final SA score to the Components, based on the Priority of their Component Types and the LoD, given the user's profile, physiological state and environment.

Priority Rules Regarding the set of rules belonging to the first category, they were defined in the following manner. The requirements analysis of the workshops led to the identification of appropriate Component Types for each *DARLENE* use case/task. Then, a questionnaire was handed out to LEAs from agencies of 5 countries, that described these Component Types and the different tasks, and requested a total ordering of the ones they considered relevant for each of the tasks, in decreasing level of importance/usefulness, so that the lower the rank of a Component Type, the more they expect that it would enhance their SA, during execution of the task. Based on the answers to the questionnaire, for each modeled task of the LEAs (e.g. patrolling), a weak ordering of the relevant Component Types is specified (ties are allowed), with at most 10 ranks, in descending level of importance. The rank of each Component Type, from 1 to 10, represents the 'Priority' of the Components of this Component Type to be displayed in the LEA's HUD, against other Components. The lower the rank and the number corresponding to the Priority, the higher the Priority is. The rules defining this Priority have the following template:

> "User(?u), Component(?c), hasTask(?u,**Task**), hasComponent(?u,?c), hasComponentType(?c,**Component Type**) -> hasPriority(**Component Type**,**Priority**)"

Where for each Task and Component Type, the corresponding Priority is set. The complete set of rules that belong to this first category that specify the Priority of the Component Types are provided in Table 3.6.

[3]https://www.w3.org/2001/sw/wiki/Pellet

Table 3.6: SWRL Rules for Priority, based on Component Type and Task

Priority SWRL Rules
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,Alerts) -> hasPriority(Alerts,1)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,Foe) -> hasPriority(Foe,2)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,Suspect) -> hasPriority(Suspect,3)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,Ally) -> hasPriority(Ally,4)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,CarriedWeapons) -> hasPriority(CarriedWeapons,5)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,CriminalActivities) -> hasPriority(CriminalActivities,6)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,UnattendedObject) -> hasPriority(UnattendedObject,7)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,AbnormalBehaviour) -> hasPriority(AbnormalBehaviour,7)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,ObjectIdentity) -> hasPriority(ObjectIdentity,8)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,Map) -> hasPriority(Map,9)
User(?u), Component(?c), hasTask(?u,Patrol), hasComponent(?u,?c), hasComponentType(?c,SummativeSceneInfo) -> hasPriority(SummativeSceneInfo,10)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Foe) -> hasPriority(Foe,1)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,CarriedWeapons) -> hasPriority(CarriedWeapons,2)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Victim) -> hasPriority(Victim,3)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,CriminalActivities) -> hasPriority(CriminalActivities,3)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Alerts) -> hasPriority(Alerts,4)

Continued on next page

Table 3.6 – *Continued from previous page*

Priority SWRL Rules
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,UnattendedObject) -> hasPriority(UnattendedObject,5)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Suspect) -> hasPriority(Suspect,5)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,ObjectIdentity) -> hasPriority(ObjectIdentity,6)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,AbnormalBehaviour) -> hasPriority(AbnormalBehaviour,6)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Ally) -> hasPriority(Ally,7)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,HealthStatus) -> hasPriority(HealthStatus,8)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,HealthRecord) -> hasPriority(HealthRecord,8)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,ProceduralInfo) -> hasPriority(ProceduralInfo,9)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,SummativeSceneInfo) -> hasPriority(SummativeSceneInfo,10)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Physiosignals) -> hasPriority(Physiosignals,10)
User(?u), Component(?c), hasTask(?u,IncidentResolution), hasComponent(?u,?c), hasComponentType(?c,Map) -> hasPriority(Map,10)

SA Rules The set of rules belonging to the second category, which assigns the SA score to the Components, were defined in the following manner. The requirements analysis of the workshops highlighted the need for the presentation and amount of information provided to the LEA's to depend on the context of use and especially the physiological state of the users. Findings of the workshops indicated that in situations of high stress or high mental workload, the field of view of the LEAs should be obscured as little as possible. Furthermore, LEA's with high expertise in the field often require less information during their operations. Based

on those insights, we defined a total ordering of the possible LoDs, for the different values of context factors modeled in our Ontology. This ordering for the currently supported context factors and values is presented in Table 3.7. It specifies the order of usefulness for each LoD, with respect to the others, in decreasing order. For instance, in the case of "High Crowdedness" and "High Stress", the "Low" LoD is favored, to avoid information overload and to minimize obstruction of field of view, whereas in the case of "Low Crowdedness", "Low Stress" and "Moderate Expertise", the "High" LoD is favored, to empower the LEA with as much detailed information as possible.

Table 3.7: Order of usefulness of LoDs based on context factors

Low Crowdedness		
	Low Stress	High Stress
Moderate Expertise	High LoD, Mid LoD, Low LoD	Mid LoD, Low LoD, High LoD
High Expertise	Mid LoD, High LoD, Low LoD	Mid LoD, Low LoD, High LoD
High Crowdedness		
	Low Stress	High Stress
Moderate Expertise	Mid LoD, Low LoD, High LoD	Low LoD, Mid LoD, High LoD
High Expertise	Low LoD, Mid LoD, High LoD	Low LoD, Mid LoD, High LoD

Based on this ordering of the LoDs, and the Priority of the Component Types, the set of rules for computing the SA score of the Components were defined given the context. Their purpose is to associate higher SA scores with Component Types of higher Priority (lower priority rank) and LoDs of lower rank in the ordering. In particular, for any Component, the higher the Priority of its Component Type, the higher its SA score is. Moreover, for Components of a specific Component Type, the lower the rank their LoD has in the ordering, the higher their SA score is. In this way, the SA score of a Component depends primarily on its Component Type and secondarily on its LoD, since Components whose Component Type has a higher Priority will always have a higher SA regardless of LoD. To achieve this, we defined the SA score of a Component to be a float of two decimal places in [0.00, 1.00). The Priority of its Component Type determines the first decimal place, while the rank of its LoD determines the second decimal place. More specifically, the first decimal place is computed as:

1 – 0.1* Priority (1)

where Priority takes values in [1, 10], with 1 to be the highest and 10 the lowest values. The second decimal place takes one of the values 0.09, 0.05, 0.01, based on the ordering of the LoDs and the LoD of the Component. In particular,

Components with the most appropriate LoD for the current context (first in the ordering) have the value of 0.09 while the ones with the least appropriate (last in the ordering) have the value of 0.01. This method for computing the SA score is encompassed by the rules presented in Table 3.7. These adhere to two different templates. The first template, used in the case of Low stress, is more generic, taking into account the LEA's environment, state, as well as profile. It is the following:

> User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,**Crowdedness**), hasStress(?s,**LowStress**), hasExpertise(?p,**Expertise**), hasLoD(?c,**LoD**), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,**LoDSA**,?s2) -> hasSA(?c,?s)

Where for LowStress and each possible value of Crowdedness, Expertise and LoD, the LoDSA, which is the appropriate value from the set 0.09, 0.05, 0.01, is added to (1) to produce the SA. In the case of High Stress, the rules become simpler, since the Expertise of the LEA isn't taken into account, as can be deduced by Table 3.8. So the template becomes:

> User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,**Crowdedness**), hasStress(?s,**HighStress**), hasLoD(?c,%s), hasPriority (?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,**LoDSA**,?s2) -> hasSA(?c,?s)

Where for HighStress and each possible value of Crowdedness, and LoD, the LoDSA, which is the appropriate value from the set 0.09, 0.05, 0.01, is added to (1) to produce the SA.

Table 3.8: SWRL Rules for SA score.

SA SWRL Rules
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,HighStress), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,HighStress), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)

Continued on next page

Table 3.8 – *Continued from previous page*

SA SWRL Rules
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,HighStress), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,HighStress), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,HighStress), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)
User(?u), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,HighStress), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)

Continued on next page

Table 3.8 – *Continued from previous page*

SA SWRL Rules
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,LowCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)

Continued on next page

Table 3.8 – *Continued from previous page*

SA SWRL Rules
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,ModerateExpertise), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,LowLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.09,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,MidLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.05,?s2) -> hasSA(?c,?s)
User(?u), Profile(?p), State(?s), Environment(?e), Component(?c), ComponentType(?cT), hasProfile(?u,?p), hasState(?u,?s), hasEnvironment(?u,?e), hasComponent(?u,?c), hasComponentType(?c,?cT), hasCrowdedness(?e,HighCrowdedness), hasStress(?s,LowStress), hasExpertise(?p,HighExpertise), hasLoD(?c,HighLoD), hasPriority(?cT,?r), multiply(?s1,-0.1,?r), add(?s2,1,?s1), add(?s,0.01,?s2) -> hasSA(?c,?s)

Based on these Rules, the Ontology Reasoner assigns the appropriate SA scores to the supported Components modeled in the Ontology, based on the current context. These SA scores are provided as input to the Optimizer so that it can reach optimal decisions, based on our modeled knowledge, in terms of which Component Instances to select for the adaptation of the UI.

3.4 UI Optimizer

In the UI Optimizer, a Combinatorial Optimization problem is formulated, with the purpose of computing the optimal UI for the display of the user at run-time. This optimal UI is the one that maximizes the SA associated with the UI, based on the modeling of our domain, while satisfying at the same time visualization and placement constrains. In particular, the UI Optimizer solves two distinct but interrelated problems at once, one of Information Element (content) and Component

(design) selection and one of Component placement (layout). On the one hand, it determines *what* information to present to end-users and *how*, which translates to the problem of selecting the appropriate Information Elements provided from the KB and the most suitable Components to visualize them. On the other hand, it determines *where* to present them, and more specifically in which of the dynamically defined possible positions in the display. The solution of the optimization problem is sent to the Visualizer module, responsible for visualizing the appropriate Component Instances, which are the selected Components at their specified position in the HUD, instantiated with the corresponding content (Information Element) from the KB.

Figure 3.6: A display grid with Sampling Rate 60px

3.4.1 Display Grid

To solve the problem of Component placement in the display efficiently at run-time, a display grid is defined, which partitions the pixels of the display into disjoint grid tiles, as we can see in Figure 3.6. The dimensions of the display grid are determined by the resolution of the display, and in particular, by the "Width Resolution" and "Height Resolution" properties, whereas the dimensions of each grid tile are determined by a configuration parameter, represented by the "Sampling Rate" property of the Ontology model. These grid tiles downsample the display, so that computations, such as positioning and collision detection are carried out more efficiently, in terms of grid tiles and not pixels. This "Sampling Rate" that determines the downsampling can vary, depending on the application and the computational resources available, so that the more computational power we have and the lower the Visualizer's rendering frame rate is, the lower its value can be, leading to more

fine-grained placement of Annotation Components with respect to their detections. In the best case, the "Sampling Rate" equals to 1, and the grid tiles are the individual pixels. Placing a Component in the display corresponds to visualizing it in the display grid, inside a block of unoccupied grid tiles, where its dimensions fit. This block of grid tiles constitutes the position of the Component, which should not overlap with that of other components. This position of a Component depends on its Component Type (e.g. “Carried Weapons”), its LoD (the higher the level, the greater the size), and its category (“Annotation”, “Detection”, or “General”). More specifically, the number of occupied tiles and the shape of the tile block is determined by the Component Type and the LoD, whereas the possible locations of the tile block in the display is determined by the category, as explained in section 3.4.2.

3.4.2 Input Sources

To define the parameters of the optimization problem, the UI Optimizer receives input from the following sources: The Ontology Model, the SA Reasoner, and the KB module.

```
{
    "type": "Foe",
    "id": 0,
    "keypoints": [
        746.75,
        314.71,
        823.96,
        314.71,
        823.96,
        437.40,
        746.75,
        437.40,
        746.75,
        314.71
    ]
}
```

```
{
    "type": "CriminalActivities",
    "id": 1,
    "detectionId": 0,
    "content" : {
        "name": "John Doe",
        "activity": "terrorism",
        "icon": "bomb.png"
    }
}
```

```
{
    "type": "SummativeSceneInfo",
    "id": 1,
    "content": {
        "Foes": 1,
        "Suspects": 0,
        "Allies": 0,
        "Objects": 0
    }
}
```

Figure 3.7: An example of 3 Information Elements. The one on the left corresponds to a "Detection" Component Type ("Foe") the middle one to an "Annotation" Component Type ("CriminalActivities"), and the right one to a "General" Component Type ("SummativeSceneInfo")

- The Ontology model provides the UI Optimizer with the necessary context information. More specifically, it supplies at run-time the physiological state of the user, and in particular their stress level, which is used to specify appropriate constraints on the number of visualized Components in the display, as we will see later in this section. Moreover, it provides display information and configuration parameters, namely the display resolution and the Sampling Rate. This is used to compute the aforementioned display grid, utilized in solving the Component placement problem.

- The SA Reasoner provides the UI Optimizer with necessary coefficients for the optimization problem, to be able to select the appropriate Components for display. More specifically, it supplies the SA score for each supported Component, so that the UI Optimizer can maximize the cumulative SA score associated with the displayed UI, subject to placement constraints.

- The KB module supplies the necessary intelligence regarding the current scene. This intelligence is provided in the form of Information Elements, that can be detected entities, as well as detection related and general purpose information. These supply different information to the UI Optimizer, depending on their Information Type. If figure 3.7, we can see 3 examples of Information Elements, one for each category of the Component Types. The UI Optimizer maps them to a Component Type based on their Information Type (one-to-one mapping) and decides which ones will be displayed, at what LoD, (through which Component) and at what position. In example, if the Component Type is an "Annotation", the id of the "Detection" it is referring to is provided, so that the UI Optimizer can position the former relative to the latter.

3.4.3 Optimization Parameters

Given these input sources, the parameters of the optimization problem can be defined (Table 3.9).

Table 3.9: List of parameters of the Optimization problem

Parameter	**Description**
$n \in Z^+$	Number of Information Elements
$E = (e_1, e_2, .., e_n)$	Information Elements, candidate for display
$T = (t_1, t_2, .., t_n)$	Component Types of Information Elements
$m_t \in Z+$	Number of Components of Component Type t
$C_t = (c_1, c_2, .., c_{m_t})$	Components of Component Type t
$l_{c_t} \in Z^+$	Number of possible positions of Component c_t
$P_{c_t} = (p_1, p_2, .., p_{l_{c_t}})$	Possible positions of Component c_t
$a_{c_t} \in [0, 1]$	SA score of candidate Component c_t
$N \in Z^+$	Maximum number of Components to display
$y_{p_{c_t}} \in [0, 0.009]$	The priority for position p_{c_t}

Each Information Element e is mapped to a Component Type t, so that the multiset T contains the corresponding t for each e. We note that T is a multiset, since two different Information Elements e_1, e_2 can have the same Component Type, thus having $t_1 = t_2$, and, in general, $|T| = |E| = n$.

Each Component Type t has a set of m_t Components c_t. These Components c_t correspond to design "templates" for the Component Type t, representing its content in different granularity levels (LoDs). Intuitively, for an Information Element e to be visualized in the display (as a Component Instance), it needs to instantiate one Component c_t, of its associated Component Type t.

Each Component c_t of Component Type t, has a SA score a_{c_t}, based on the current context. This SA score is provided as input by the SA Reasoner. It approximates the Component's appropriateness for the displayed UI, given the user's profile, psychological state environment and task.

Furthermore, each Component c_t can have multiple possible positions p_{c_t} in the display, whose number, denoted as l_{c_t}, and location depend both on the category of its Component Type t (e.g. Annotation, Detection, General) and its LoD (e.g. LowLoD, MidLod, HighLoD). As mentioned earlier, each possible position corresponds to a block of tiles of the display grid. In the final UI, the positions of the displayed Component Instances should not overlap, thereby each tile of the display grid can be occupied by a single instantiated Component.

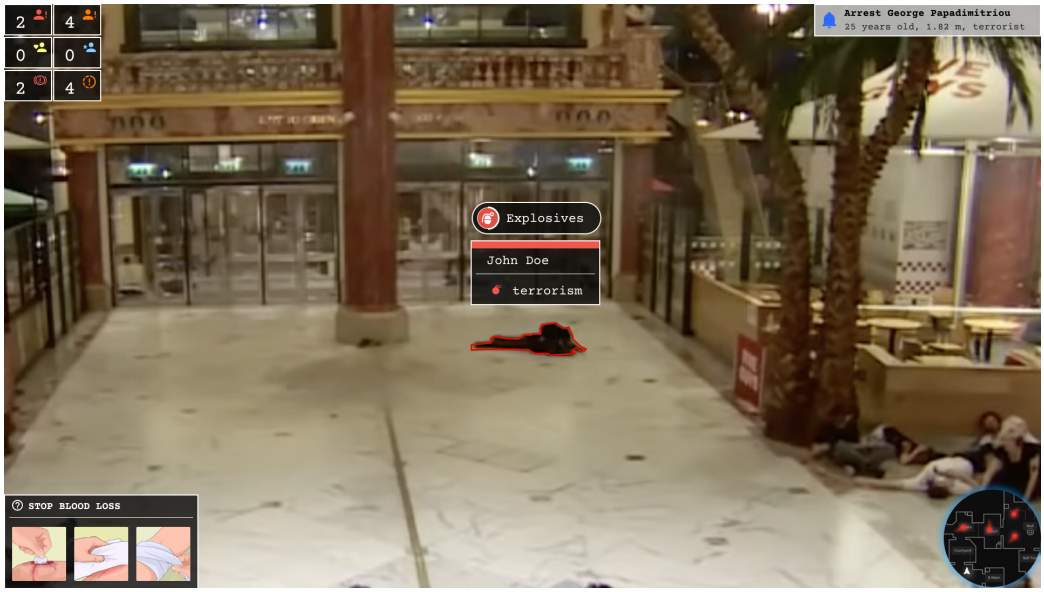

Figure 3.8: An example of possible positions for "Annotation" and "General" Components. The "General" Components are placed relative to the display, e.g. on the four corners, whereas the "Annotation" Components are placed relative to the corresponding detection

Components of "Detection" Component Type have only one possible position, which coincides with the block of grid tiles occupied by the bounding box of the detected human/object. Contrarily, Components of "Annotation" and "General" Component Type have plural possible positions. In the case of Components of "Annotation" Component Type, the possible positions are relative to the detection

they are referring to. For instance, they can be located above, to the left or to the right of the corresponding detection, or stacked, one on top of the other, as in Figure 3.8. For Components of "General" Component Type, their possible positions are relative to the display, for example, blocks of grid tiles on its four corners, as in Figure 3.8. In all cases, a Component can occupy at most one of its possible positions.

Intuitively, the purpose of multiple possible positions is to provide more flexibility to the optimization algorithm on selecting the most appropriate Component for displaying an Information Element, given the constraint that there is no overlapping. Concretely, they result to a Component having multiple 'attempts' of being visualized in the UI, increasing its chance of not colliding with a different Component which the optimization algorithm will "favor", since, for example, it is associated with a higher SA.

In order to be able to prioritize some positions over others, for different Components, (e.g. the top right corner position for the 'Alert' Component Type in High LoD, or the position above the detection, for the Annotation "Criminal Activities" in Mid LoD), we define the variable $y_{p_{c_t}}$ as the priority for position p_{c_t}. It depends both on the Component Type t and LoD presentation c and can take values in the range [0, 0.009]. Using this domain, the variables prioritizing positions $y_{p_{c_t}}$, taking values below 0.01, affect less than the situational awareness coefficients a_{c_t}, taking values of 0.01 and above, the cumulative SA that the UI Optimizer tries to optimize. This is in line with what we wish to express, namely that the actual Component is more important than its position. This $y_{p_{c_t}}$ serves multiple purposes. One primary aim is to avoid oscillations in the selected Information Elements and their positions, at run-time, and leverage spacial memory. These oscillations are caused by the multiple optimal optimization solutions, given that Information Elements of the same c_t, have the same a_{c_t}. By giving a higher $y_{p_{c_t}}$ to Information Elements selected in the previous frame, the optimization 'favours' them over others of the same a_{c_t}, and tries to place them again and in the same position.

Finally, the parameter N denotes the maximum number of Components that can be instantiated and displayed in the UI, given the current context. Concretely, in order to avoid information overload and induced stress by the UI, an IF-THEN rule based approach has been adopted, which reduces the number of possible components of the UI, based on the user's psychological state. In particular, the rule takes into consideration the LEA's stress and decreases the value of N in cases of increased stress. The number of components to display if the user has 'Low Stress' is 9, whereas in the case of 'High Stress' the number is 5. The choice of these numbers are based on "Miller's law" regarding the capacity of short term ("working") memory, which states that most adults can store there between 5 and 9 items [113].

3.4.4 Objective function

Our goal is to optimize the SA score associated with the displayed UI dynamically, by determining which of the Information Elements to display, using which Components (Component Types at a specific LoD) and at what position, at run-time. To this end, we use integer linear programming to maximize the cumulative SA score of the LEA's UI. As mentioned above, each Information Element e, is mapped to Component Type t. The binary decision variable $x_{p_{c_t}} \in 0, 1$ denotes whether the Information Element e mapped to t is displayed or not, through Component c_t, and at the position p_{c_t}. The objective function, which expresses the total SA score of the UI, is formulated as follows:

$$f(x) = \sum_{t} \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} * (a_{c_t} + y_{p_{c_t}}) \quad \textbf{(1)}$$

The optimization objective is to maximize the objective function f, by selecting the appropriate values for x. Concretely:

$$\max_{x} f(x) \quad \textit{subject to } v(x),\ c(x),\ u_t(x)\ \forall t \in \{1, \ldots, n\} \quad \textbf{(2)}$$

3.4.5 Constraints

For our purposes, maximizing the total SA score of the displayed UI is not sufficient. It should at the same time satisfy specific conditions, in order to avoid redundancy of information, information overloading and collisions between UI components. We will again map each Information element e, to its corresponding Component Type t for use in the equations. The space of feasible solutions of the optimization problem is restricted by the following constraints:

Uniqueness constraint In order to ensure that each Information Element is displayed through at most one Component, and at most in one of its possible positions, we add the following constraints, for each candidate Information Element:

$$u_t(x) = \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} \leq 1, \forall t \in \{1, \ldots, n\} \quad \textbf{(3)}$$

Visualization constraint Furthermore, we need to ensure that the number of Information Elements, supplied through displayed Component Instances in the UI, does not surpass the maximum number N, defined to avoid information overload and induced stress. To achieve that, we include the following constraint:

$$v(x) = \sum_{t} \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} \leq N \quad \textbf{(4)}$$

Collision constraint Finally, we need to ensure that there are no overlapping Component Instances in the displayed UI. This is achieved by verifying that for every pair of Component Instances, there is no collision, in terms of sharing at least one grid tile of the display grid. We define the predicate $isCollided(p1_{c1_{t1}}, p2_{c2_{t2}})$, denoting whether the Information Element of Component Type $t1$ materialized through Component $c1$, at position $p1$, collides with the Information Element of Component Type $t2$ materialized through Component $c2$, at position $p2$. Therefore, the collision constraint can be formulated as follows:

$$\forall p1_{c1_{t1}}, p2_{c2_{t2}}$$

$$x_{p1_{c1_{t1}}} * x_{p2_{c2_{t2}}} * isCollided(p1_{c1_{t1}}, p2_{c2_{t2}}) = 0,$$

$$p1_{c1_{t1}} \neq p2_{c2_{t2}} \quad \textbf{(5)}$$

As a result of this constraint, the visualization of an Information Element through a Component Instance, and in particular, whether it is displayed, through what Component, and thereby in what LoD, and in which position, is sensitive to the placement and shape of the other Component Instances for display in the UI. For instance, a Component Instance may take a sub-optimal position (of lesser priority $y_{p_{c_t}}$), due to the fact that the optimal one is occupied. Moreover, an Information Element may be displayed through a Component in a decreased LoD (more succinct information), in order to occupy less space in the UI and fit without overlapping. More importantly, another effect of this constraint could be that an Information Element is not displayed at all, because for all its the possible Component Instances (LoDs and positions), it overlaps with Component Instances materialized through Components which are associated with a higher SA score and are 'preferred' by the optimization algorithm.

Collision constraint implementation For each possible position of an Information Element, a boolean 2D array is defined which represents the display grid. This array has dimensions $\frac{WidthResolution}{SamplingRate} \times \frac{HeightResolution}{SamplingRate}$ and its elements represent the grid tiles of the display. The block of tiles of the display grid that correspond to this position are set to True, whereas all the rest are set to False.

The set of all positions of all Information Elements are stored in a boolean 3D array, for the purpose of computing collisions between positions to be occupied in the display. Its first dimension represents the candidate positions competing to be occupied in the UI, whereas the other two dimensions contain for each position the aforementioned 2D array.

To formulate programmatically the collision constraint **(5)**, we denote as *posArray* the resulting 2D array after reshaping the aforementioned boolean 3D array to flatten its last two dimensions. Consequently, the first dimension of the posArray represents the positions, whereas the second represents the tiles of the 2D display grid, but flattened to 1D. Furthermore we denote as x the 1D boolean array containing the decision variables $x_{p_{c_t}}$ for each competing position. We define the 2D

array *selectedPosArray* as the logical AND operation between *posArray* and x:

$$selectedPosArray = AND(posArray.T, x)$$

The reason for this operation is that we are interested to check if there are collisions only among positions that are selected to be displayed.

The purpose of the collision constraint is to impose that each tile is occupied by at most one Information Element, visualized through a specific Component and at a specific position. To express that programmatically, we utilize the following constraints, where gt denotes a grid tile:

$$\forall gt \quad sum(selectedPosArray(gt, :)) \leq 1$$

Where *sum(selectedPosArray(gt, :))* sums across the positions axis of tile *gt*.

Figure 3.9: Positioning of an Annotation in terms of grid tiles (left) and in terms of pixels (right). The Sampling Rate is 60px

Fine-grained vs Coarse Collision detection As already mentioned, the positions of the Component Instances are computed in terms of blocks of grid tiles. The lower the Sampling rate, the smaller the grid tiles become, leading to more fine-grained positioning. In run-time applications, this fine-grained positioning becomes even more aesthetically desirable for Annotation Components, which are placed relative to a potentially rapidly moving Detection Component, and follow it in the scene. However, the lower the sampling rate, the more computationally expensive the Collision constraint is, potentially compromising the real-time computation requirement.

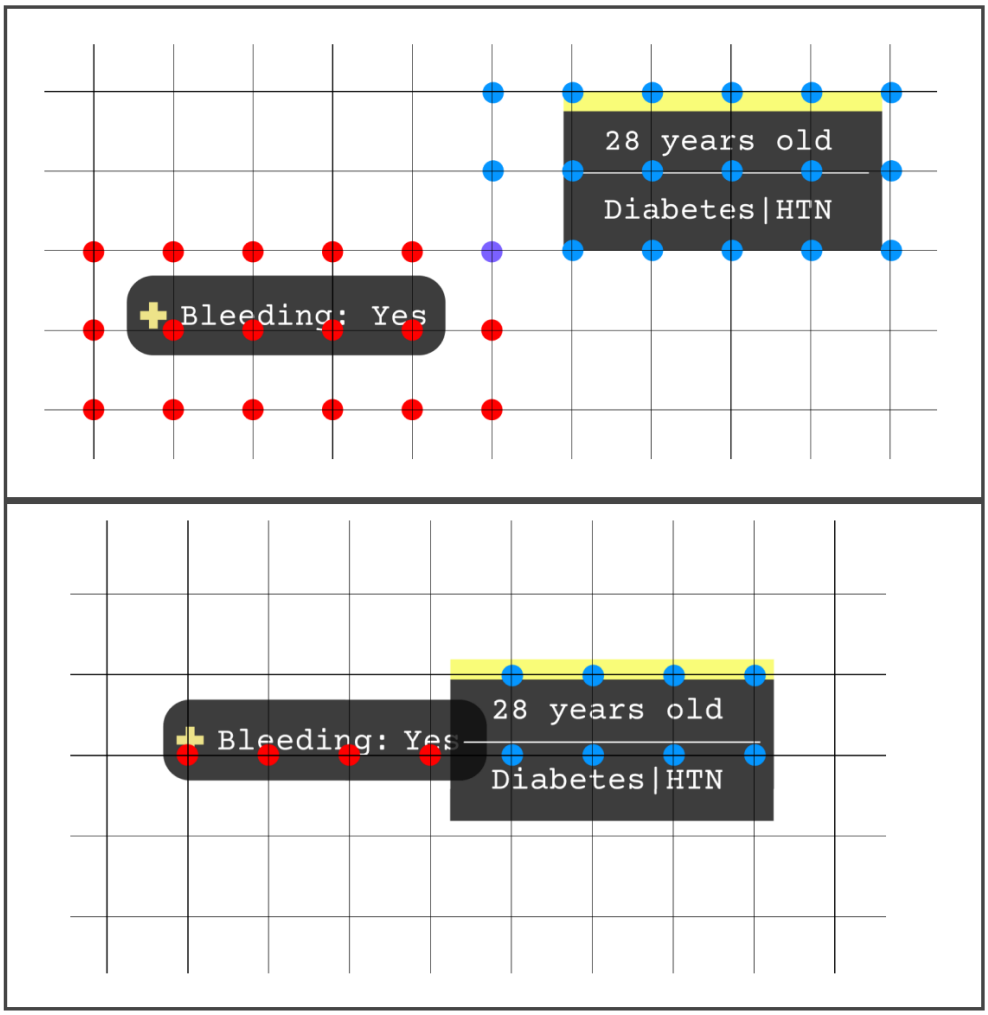

Figure 3.10: An example of the "over-sensitive" (top) and "under-sensitive" (bottom) Coarse collision approach

For an improved visual result, we can compute the position in terms of pixels (instead of grid tiles), but keep the Collision constraint in terms of grid tiles (with dimension smaller than the smallest Component), meaning that Collisions are still checked at every tile. In Figure 3.9, we can see an example of the positioning of an Annotations in both cases (grid tile positions and pixel positions). If we follow this pixel positioning approach instead of the grid tile approach, we have a Coarse collision detection instead of a fine-grained one. The coarse collision detection can be divided into two types, an "over-sensitive" and an "under-sensitive" collision detection, depending on how we map the pixel position to the grid tiles of *posArray*. In both cases, complete overlapping is ensured to be avoided, provided that the grid tile dimension is smaller than the smallest Component. In Figure 3.10 we can see the shortcomings of each approach, using corner cases. The red and blue dots correspond to tiles of the *posArray* with value *True*. In the "over-sensitive" case (top), a collision is detected (purple dot), although there is none, whereas in the "under-sensitive" case, no collision is detected although there is partial overlapping. In the context of the simulation videos of the user-based evaluation, described in 5, Coarse, "under-sensitive" collision was preferred, displaying in all cases satisfactory results (with only occasional, small partial pixel overlappings). We attribute these results to the variable run-time distance between Annotations, leading to the rule of thumb that Annotations that partially overlap (in pixels) will eventually collide (in grid tiles). As a result they change position and there is no pixel overlapping anymore.

3.4.6 Solving the optimization problem

For the purpose of solving the integer linear program defined in equations (1) – (5), the *Gurobi*[4] mathematical optimization solver is used. An initial implementation with the *CPLEX* [5] optimizer wasn't able to define the collision constraint efficiently, due to the optimizer's lack in expressive power. More specifically, the efficiency of Matrix operations that include decision variables can not be leveraged, since their values are determined only at run-time, and not during the formulation of the optimization problem. This issue was overcome in the implementation with Gurobi, which allows some Matrix operations with still uninitialized decision variables. However, improved formulations that, for instance, utilized the decision variables as indices to "trim" the *posArray*, are not supported. Thus, we use the *selectedPosArray* for collision computation, summing across all positions.

Since the formulated optimization problem is a variation of the knapsack problem, which is known to be NP-hard, we tested our approach with realistic examples, to verify that it can be solved in real-time. In the Expert-based evaluation we conducted, detailed in Chapter 4, the DM was run with display resolution 1920x1080 and Sampling rate 60px. The number of LoDs for each Component were 3 and the

[4]https://www.gurobi.com/
[5]https://www.ibm.com/analytics/cplex-optimizer

possible positions for each one of their LoDs was equal to 3. The Information Elements provided by the KB were approximately 9-15, depending on the scenario. To this end, a commodity gaming PC (Intel Core i7-8086K, 4GHz with 6 cores, 32GB Ram, Windows 10, NVIDIA GeForce GTX 1080 Ti) was utilized. On average, it took 0.0178 seconds with standard deviation 0.0015, more than 2 times less that the 0.04 seconds requirement for our application. In the User-based evaluation we conducted, detailed in Chapter 5, the DM was run using the same specs, with the difference that we had 9 positions for each of 2 possible LoDs. In all simulation videos, the DM required < 0.02 seconds for each frame, giving again real-time results.

3.5 Stress Detection

A key context factor that the DM takes into account for the *DARLENE* use case is the Stress level of the LEA. The impact that Stress has on SA is highlighted in [47], were it is portrayed as an "SA Daemon", constituting a prominent factor that can negatively influence it. This effect was also confirmed in the Co-creation Workshops, which identified as a requirement that the system should accommodate the stress condition of the LEAs. To this end, a stress-detection and a stress-classification model were developed as a core unit of the Context module, providing at run-time the stress state of the LEA to the DM.

In the field of stress analysis, the electrocardiogram (ECG) signal has proven to be a reliable, non-invasive physiological measure for stress detection and classification [66]. In recent years, stress models based on Deep Neural Networks (DNNs), although still at early stages of research, have produced promising results [138]. In contrast to traditional Machine Learning (ML) techniques, which require handcrafted features, Neural Networks are able to learn appropriate features automatically from raw physiological signals. Furthermore, Convolutional Neural Networks (CNNs), which have become the de facto standard in several image classification applications, have recently been proposed and have achieved state-of-the-art performance levels when applied to 1D signals, such as physiological signals. [99].

Taking the above into consideration, we have proceeded with the construction and assessment of CNN architectures for stress level classification through the analysis of ECG signals, as detailed in Tzevelekakis, Stefanidi and Margetis. [148]. This work pursued the implementation of ML architectures with reduced memory footprint, having as a requirement to fit in the LEAs' wearable devices, which employ micro-computers with limited CPU and memory capacity (e.g., NVIDIA jetson). In the context of this Thesis, one of the specified CNN architectures was developed, using ultra short-term raw ECG signals (3 seconds), for a stress detection model (*Stress*, *no Stress*), and a stress classification model, which classifies stress into 3 classes (*Low*, *Moderate*, *High*). The resulting models were trained and evaluated on the Stress Recognition in Automobile Drivers dataset (DriveDB), achieving state-of-the-art performance in comparison to similar DNN approaches.

In the following sections, we provide a brief description of the aforementioned work. Initially, we elaborate on the Dataset we have utilized, as well as pre-processing and data augmentation techniques that we have applied. Then, we describe the proposed architecture, as well as the results of cross-validation, and compare them with existing approaches.

3.5.0.1 Dataset

The Stress Recognition in Automobile Drivers dataset (DriveDB) is the most prevalent, publicly available ECG stress-related dataset [77] for training ML models. It contains a collection of multi-parameter recordings from 17 healthy volunteers, taken while they were driving on a prescribed route including city streets and highways in and around Boston, Massachusetts, for more than an hour. The annotations of the driving periods were based on three major conditions of the experiment: resting periods that were assumed to be of *Low Stress*, highway driving periods of *Moderate Stress*, and city driving periods of *High Stress*. The recorded signals include the ECG, the Electromyography (EMG) (right trapezius) signal, the Galvanic Skin Resistance (GSR), measured on the hand and foot, and the respiration signal. To capture the ECG signal, it follows the lead II standard configuration, minimizing motion artefacts and producing a rhythm trace with sharp R-waves. From the aforementioned signals, only the ECG signal is utilized in the context of this work.

Pre-processing In order to convert the dataset to a label-sample form, a marker signal embedded in the ECG signals was leveraged. This marker signal has distinguishable peaks that separate ECG signal segments which are annotated with different stress states. To apply the annotations dictated by this marker signal, we up-sampled it from its initial frequency of 15.5Hz, to 496Hz, to match the frequency of the ECG signal, and then located its peaks. For parsing and re-sampling the data, the waveform database (WFDB) software package [114] was used, whereas for finding the peaks, the SciPy[6] python package was utilized.

Errors and problems with the ECG and marker signals of some subjects, such as missing peaks, led to the elimination of specific drives or driving segments, as performed in other works e.g [8]. As a result, data from 9 participants were utilized. Moreover, following the approach of the dataset's authors [78] [77], for each driving segment, corresponding to one of the experimental conditions, the middle 5 minutes were extracted and utilized, to avoid any marginal signal noise that is located at the borders of each segment. Finally, similar to the works [77] and [66], we performed a baseline normalisation procedure, with the purpose of reducing the individual bias, introduced in the measurements as a result of the differences in age, posture, level of physical conditioning, breathing frequency and other factors.

[6] https://scipy.org/

Data augmentation Considering the relatively small size of the dataset for effectively training a DNN, we employed a data augmentation technique to generate more training data. In particular, we followed a sliding window approach, similar to ones employed for 1D signals in other application domains e.g. [4]. To this end, training samples were generated from multiple window-sized crops of the an initial 1D signal segment, using a pre-defined stride.

3.5.0.2 Architecture

The CNN architecture that we developed increases the number of channels, as the input dimensionality decreases in deeper layers, similar to the well-established VGG architecture, widely used for efficient large-scale image recognition [139]. In particular, the number of channels starts from 64 and then increases by a factor of 2 at each stage, until it reaches 512. Our architecture includes 5 stages of 1D layers, with each stage consisting of the following layers: Convolution, Batch-Normalisation, Activation, Max-Pooling and Drop-Out. The Batch Normalisation layers to each stage, along with the Drop-Out and leaky RELU activation layers reduce overfitting and minimize the generalisation error. After these 5 stages, a Global Average Pooling layer and a Fully-connected layer conclude the model. Figure 3.11 illustrates the described architecture.

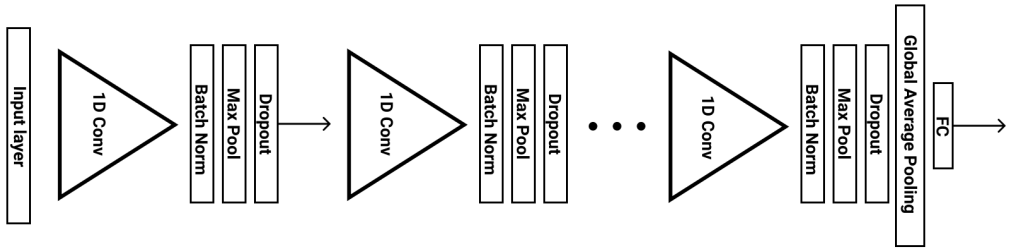

Figure 3.11: Model architecture

3.5.0.3 Results

The models for both stress detection and 3-level stress classification were trained and evaluated on the pre-processed dataset originating from the DriveDB. With regard to the stress-detection model (2-level classification), the 'High' and 'Moderate' stress classes of the dataset were mapped to the 'Stress' label, whereas the 'Low' class was mapped to the 'no Stress' label. For assessing their performance, we employed 4-fold cross-validation, using the 25% of the data as a validation set each time. We adopted the approach followed by Seo et al. [137] and the partitioning into folds was carried out across subjects. More specifically, each fold can contain signal segments of different drivers, and each driver can have segments in multiple folds, having shuffled the collection of all driver segments before the split.

We tested different hyper-parameters as well as configurations with regard to window size and stride and used the Adam Optimizer[98] for minimizing the loss of the objective function.

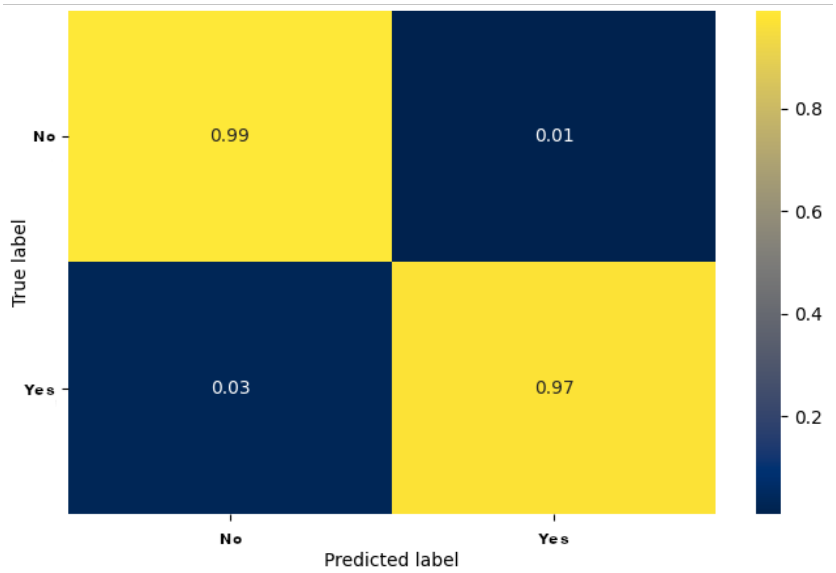

Figure 3.12: Confusion matrix of model for the stress detection task

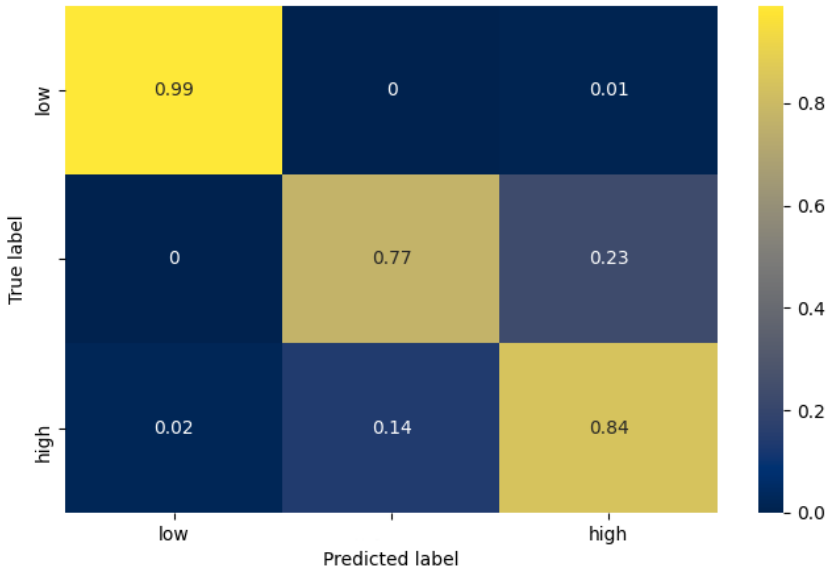

Figure 3.13: Confusion matrix of model for the 3-level stress classification task

Our model achieved an accuracy of 0.853 ± 0.007 for the 3-level classification task, and 0.986 ± 0.002 for the stress detection task, achieving state-of-the art

performance. In Figures 3.12 and 3.13, we can see the confusion matrices for the two tasks. In the case of 2 classes (stress detection), we can see that the true positive and true negative rate is very high (97%, and 99% respectively), while the false negative rate is 3 times higher than the false positive rate. Furthermore, in the case of 3-level stress classification, we can observe that, as expected, the model can differentiate with higher accuracy between the low and high classes, than between the moderate and high classes.

Table 3.10: 3-level Stress classification on DriveDB

Models	**DeepERNet**	**He et al.**	**DeepECGNet**	**Our Model**
Accuracy (%)	83.9	82.7	80.7	85.3
Frequency (Hz)	496	496	496	496
Time Window (sec)	50	50	50	3
Augmentation	no	no	no	yes
Signals	ECG&RSP	ECG	ECG	ECG

For DeepERNet refer to [137], for DeepECGNet refer to [90], for He et al. model refer to [76].

We first compare our model's performance on the 3-level classification task, with that of 3 other DNNs, evaluated on the DriveDB dataset in the work of Seo et al. [137]. These models are the state-of-the-art DNN models on the DriveDB dataset, and among the best-performing for 2- and 3-class stress classification, using heart related features. In Table 3.10, we can see that our models' accuracy of 85.3% surpasses that of the other models, while using a smaller time window as input for making a prediction. Moreover, our approach achieves its accuracy while utilizing only one signal, the ECG signal, in comparison to [137], which also uses RSP.

Table 3.11: Stress detection using heart related features

Method	**Accuracy (%)**	**Method**	**Data**	**Window size**
[73]	98.69	CNN	HRF	10s
Our Model	98.6	CNN	ECG	3s
[93]	98.3	CNN-LSTM	ECG	-
[101]	95.67	CNN	HR & other	30s
[36]	90.19	CNN	ECG	10s
[66]	89.8	CNN	ECG	60s
[87]	87.39	CNN-RNN	ECG	10s
[76]	82.7	CNN	ECG	10s

In Table 3.11, for the stress detection task, we compare our results with those of other works which use heart related features, independently of the evaluation dataset choice. We observe that our model has only 0.09% difference from [73], while using a 3s time window instead of 10s, and automatically learned representations instead of handcrafted ones.

Chapter 4

Expert Based Evaluation

As a first qualitative assessment of our computational approach, an expert-based evaluation was conducted, involving 10 experts: 5 UX and 5 LEA experts. In general, an expert-based evaluation is recommended as a means to ensure that, prior to testing a product with actual end-users, a considerable number of problems, which can be identified through other methods, has been eliminated. They are suggested to be used early in the development life-cycle, and in complementarity with user testing [84]. The aim of this preliminary evaluation was to assess the decisions of the DM, regarding Component selection (what), Component LoD (how) and Component placement (where), given a context instance. To this end, different scenarios were created, encompassing different experimental conditions. For each experimental condition, our algorithm was run on appropriate images, generating a User Interface. In particular, the conditions were concerned with the LEA's expertise (high or moderate), stress (high or low) and task (patrolling or incident resolution), as well as the crowdedness of the environment (low, high). For the purpose of this evaluation we did not examine low LEA expertise, since this condition is highly unlikely in realistic conditions, given that officers are well trained before being engaged in incident resolution or patrolling tasks. In Table 4.1, the 12 conditions of the evaluation are presented.

Table 4.1: The Experimental Conditions of the preliminary evaluation

	Crowdedness	
User Conditions	Low	High
High Stress	Patrolling Incident Resolution	Patrolling Incident Resolution
Low Stress, High Expertise	Patrolling Incident Resolution	Patrolling Incident Resolution
Low Stress, Moderate Expertise	Patrolling Incident Resolution	Patrolling Incident Resolution

Figure 4.1: Selected Images for the Scenarios of the Evaluation: (top-left) Patrolling - Low Crowdedness, (top-right) Patrolling - High Crowdedness, (bottom-left) Incident Resolution - Low Crowdedness, (bottom-right) Incident Resolution - High Crowdedness

4.1 Procedure

In detail, our procedure was as follows. Before the study, the Decision Making module was run for each condition on appropriate images, given the context. In particular, 4 scenarios were created, one for each combination of task and crowdedness level, where different persons and objects of interest are present in the scene. A representative image was selected, depending on the task, crowdedness level and detected persons (e.g. suspects and victims), and objects (e.g. unattended Object of the scenario). In Figure 4.1, we can see the selected images for each scenario. Then, for every scenario, the algorithm run 3 times for the 3 user conditions of Stress and Expertise, producing the UIs for the 12 conditions of the experiment. In Table 4.2, the user conditions for each of the scenarios are listed. In Figure 4.1, we can see an example of a generated UI, for Scenario B, for the condition with Low Stress and Moderate Expertise.

The study was conducted as an online session using a teleconferencing platform, due to the ongoing Covid-19 pandemic. Participants were first explained the main parameters which affect the decision making. More specifically, they were introduced to the context factors taken into account for the generation of the UI. Then, they were introduced to the supported GUI Components, familiarizing themselves

Figure 4.2: The Generated UI for Scenario B: Patrolling - High Crowdedness

with the different Component types and their variations with regard to the LoD. After this introductory phase, the main part of the experiment began. For each of the scenarios, the participants were first introduced to the simulated situation (e.g. there are two Foes, one of them armed, when the LEA is patrolling in a crowded area). Then, the generated UIs for the different user conditions where shown. The experimental conditions were counterbalanced and randomly assigned to participants, so that each participant would examine conditions in a different order, aiming to counterbalance carryover effects [27].

Table 4.2: The 4 Scenarios and the User conditions

Scenarios	**User Conditions**		
A. Patrolling Low Crowdedness	High Stress	Low Stress, High Expertise	Low Stress, Moderate Expertise
B. Patrolling High Crowdedness	High Stress	Low Stress, High Expertise	Low Stress, Moderate Expertise
C. Incident Resolution Low Crowdedness	High Stress	Low Stress, High Expertise	Low Stress, Moderate Expertise
D. Incident Resolution High Crowdedness	High Stress	Low Stress, High Expertise	Low Stress, Moderate Expertise

For each of the UIs, the participants were asked to assess the appropriateness

of the visualization decisions of the algorithm, given the conditions. In particular, they were asked the following questions:

- Do you think that the components visualized are appropriate, for the current context?
- Do you think that the LoD for each component is appropriate?
- Do you think that placement of components is appropriate?

In answering these questions, they were prompted to elaborate as much as they could and to try to justify their views. To have a quantitative indication of their satisfaction in relation to the three aspects assessed, we used a modified version of the Success Rate metric [51] in which Success (S) corresponded to participants responding positively to a question, Partial Success (PS) to responding positively but identifying points of improvement, and Failure (F) to responding negatively. In particular, two independent evaluators gave a score for each question, and any differences in the assigned scores were then discussed via a consensus building approach. In the cases of PS and F scores, participants' comments and feedback were also noted, in order to guide future improvements. The formula of the Success Rate has as follows:

$$SuccessRate = N_S + 0.5 * \frac{N_{PS}}{N}$$

Where N_S is the number of answers scored as a success, N_{PS} the number of "Partial Success" answers and N the total number of responses. At the end of the evaluation, the participants were asked to provide general comments and suggestions, and were debriefed.

4.2 Results

In Table 4.3 the Success Rate scores of each of the experimental conditions is presented. In the following sections, the resulting scores for each of the evaluation pillars, namely Component selection (what), Component LoD (how) and Component placement (where), will be interpreted, and analyzed. Moreover, possible future directions are presented for addressing some of the identified limitations.

Table 4.3: Success Rate scores for each experimental condition

Scenarios	**Scores**	**High Stress**	**Low Stress High Expertise**	**Low Stress Moderate Expertise**
A. Patrolling Low Crowdedness	Selection	0.65	0.95	0.95
	LoD	0.95	0.8	0.7
	Placement	1	0.95	0.8
B. Patrolling High Crowdedness	Selection	0.6	0.95	0.85
	LoD	0.9	0.75	0.9
	Placement	1	0.95	0.65
C. Incident Resolution Low Crowdedness	Selection	0.65	0.95	1
	LoD	1	0.95	0.5
	Placement	0.6	0.95	0.5
D. Incident Resolution High Crowdedness	Selection	0.6	0.75	0.75
	LoD	0.95	0.85	0.65
	Placement	1	1	0.75

4.2.1 Component Selection

The choice of what components to visualize received a high score and was appraised in the conditions involving Low Stress (range [0.75, 1], mean = 0.893, std = 0.09, median = 0.95). On the contrary, in the case of High Stress, the component selection decision received poor reviews across the rest of the conditions, with little score deviation (range [0.60, 0.65], mean = 0.625, std = 0.025, median = 0.625). In particular, the participants agreed that the algorithm sometimes wasn't displaying all the necessary information based on the scenario. For example, in some scenarios, some unattended objects weren't highlighted. This is due to the stricter constraint of the UI optimizer on the number of visualized components in case of High Stress, than in the case of Low stress. Many participants expressed the view that all detections should be highlighted. In addition, in the scenarios of Incident Resolution (scenarios C, D), in the case of High Stress, there was disagreement on whether to show some Components. Approximately half of the participants, primarily LEAs, suggested that minimum information is shown ("only the threat matters in such a situation"), whereas the other half, primarily UX experts, suggested that more information, e.g. regarding victims, is displayed, but in low LoD.

Regarding the problem of not displaying all the necessary components in the High Stress condition, the Visualization Constraint of the UI Optimizer needs to be updated. Based on the findings of the evaluation, one solution would be that the Detection components, visualized as highlights, aren't counted against the maximum number of components that can be displayed. Another, perhaps improved,

solution would be to restrict the percentage of occluded field of view by the visualized Components, depending on the context, instead of the number of visualized Components. The Detection Components, being mere highlights wouldn't contribute. With respect to the disagreement of participants on whether to show only the most essential Components in critical situations such as Incident resolution, personalization aspects could be incorporated to address the difference in preferences. For example, the SA associated with the different Component can be customized for each user, instead of having global values.

4.2.2 Component LoD

The choice of how to present the Components Instances, and in particular in which LoD, received a high score and positive comments, in the conditions involving High Stress or Low Stress with High Expertise (range [0.75, 1], mean = 0.893, std = 0.08, median = 0.925). However, this score dropped in the condition involving Low Stress and Moderate Expertise (range [0.5, 0.9], mean = 0.6875, std = 0.14, median = 0.675). A primary cause for this, verified by the participants' feedback, is the choice to prioritize the display of Components in High LoD, resulting to the occlusion of important parts of the scene. This choice was especially criticized in the "Incident Resolution" conditions (scores 0.50, 0.65), where occluding the LEAs field of view was considered particularly problematic.

To address the problem of occlusion of Components in High LoD, a suggestion by a participant was to automatically decrease their LoD after a few seconds.

4.2.3 Component Placement

Regarding the choice of where to place the Components Instances, in the conditions of High Stress, and Low Stress with High Expertise, with the exception of scenario C (score 0.60), the UIs generally received a high score and positive feedback (range [0.95, 1], mean = 0.979, std = 0.025, median = 1). In the case of Low Stress with Moderate Expertise, the score drops considerably (range [0.50, 0.80], mean = 0.675, std = 0.115, median = 0.7). In general, the problems identified related to the Annotation Components. The placement of General Components received positive feedback and the highlighting of the detections is standard. In some cases, like the aforementioned exception in scenario C, in the High Stress condition, participants reported that it was confusing, to whom an annotation was referring to. This problem was particularly prominent when the annotation was displayed lateral to the detection and there were civilians next to the detection. Since its possible that Annotation Components are displayed for a detection without it being highlighted, it was sometimes mistakenly perceived that the annotations belonged to a civilian, who was perceived as a person of interest e.g. Foe. Furthermore, although the relative position of annotations was standard across conditions (above, left and right of the corresponding detection), in the Low Stress with Moderate Expertise conditions, the choice to prioritize the High LoD led to Components of bigger size.

As a result, they were occluding important information in the scene, e.g. civilians, being placed on top of them. A reason for this is that collision checking and avoidance applies only among detected persons and objects, and Annotations.

To address the problem of ambiguity, regarding whom annotations belong to, a first step would be to always highlight detected objects and humans that have at least one Annotation visualized. Another potential improvement would be to have more possible positions for Components on top of the corresponding detections, since participants seemed to favor them. In particular, some participants suggested to display the different Annotation Components of a detection stacked, the one on top of the other, above the detection. Regarding the problem of occlusion, caused by Component Placement, an approach that would limit it could be receiving as input detections of humans that are not persons of interest e.g. civilians, just so that they will be considered for collision avoidance with Components.

4.3 Conclusion

All in all, this preliminary expert evaluation led to important insights regarding the DM's decisions, in particular, what Components to visualize, how – in which LoD, and where in the display. It identified limitations and directions for improvement as well as strengths. In general, all participants found the system very promising, having the potential to support LEA's during their operations. On the one hand, the LEAs emphasized more on operational issues, suggesting that the agent's field of view should be as clear as possible, even in no-stress situations. On the other hand, UX experts emphasized on usability aspects, giving useful suggestions. Certain inconsistencies were noted in preferences across participants, which need to be further explored. Further field studies with larger participant samples were performed, as reported in the next section, to identify additional issues and explore the potential improvements introduced for each assessed aspect.

4.4 Improvements

The insights acquired from this expert-based evaluation guided improvements in the DM, before continuing to a subsequent user-based evaluation. A notable modification that was carried out, based on the findings analysed in the previous sections, was that the Detection components, visualized as highlights, are always displayed. As a result, collisions among detections are not taken into account. However, detection collisions with any other Annotation Component or General Component are avoided, with the detection prevailing and being displayed. Moreover, another improvement was the incorporation of more possible positions for Components and the suggestion for "stacking", when possible.

Chapter 5

User Based Evaluation

To explore further benefits and limitations of our computational approach and investigate whether it leads to an enhanced Situational Awareness (SA) in our LEAs application domain, we conducted a User-Based evaluation with 20 police officers. The evaluation was performed in the form of an XR simulation, replicating real scenarios in a reproducible and controllable way, while avoiding the risks of performing them in reality [39, 54]. In particular, during the experiment, the participants watched, in an AR HMD, videos portraying staged terrorist attacks in different experimental conditions. Through these videos, their SA was measured using the Situation Awareness Global Assessment Technique (SAGAT) query technique [50], answering to questions which evaluated their perception of the situation at arbitrary instants. Standardized Questionnaires were also utilized, to estimate their subjective SA, their mental workload and their UX. Participants performed the task with and without the system enabled. Moreover, given that the mental state of the user, and in particular their Stress state, is a key context factor for our use case, the task was performed both with and without experimentally induced stress. With respect to our research questions, we aimed to assess 1) whether our approach enhances SA, 2) the mental workload induced by the system, and 3) the total User Experience of the system, both in normal physiological state and under stress. In Figures 5.1 and 5.2, frames from two different simulation videos are displayed, in non-stressful and stressful conditions, respectively.

Figure 5.1: A frame of one of the simulation videos, in a non-stress condition

Figure 5.2: A frame of one of the simulation videos, in a stress condition

5.1 Hypotheses

In particular, our hypotheses were the following:

H1a. The system enhances situational awareness in stress conditions

H1b. The system enhances situational awareness in non-stress conditions

H2a. The system does not impose workload to the user in stress conditions

H2b. The system does not impose workload to the user in non-stress conditions

H3a. The overall UX of the system is positive when the user is stressed

H3b. The overall UX of the system is positive when the user is not stressed

5.2 Methodology

The following sections will describe the methodology that was followed for designing of the study.

5.2.1 Experimental design

Regarding our experimental design, we used a within-subject design with two independent variables, namely the Stress state of the participant, taking values *Stress* and *no Stress*, and whether our computational approach was used or not, taking values *with System* and *without System*, yielding 4 conditions under which the simulation videos were shown. The order of the conditions was randomized across participants, adopting a 4x4 Latin square design and assigning the simulated scenarios to the following conditions *with System-Stress*, *without System-Stress*, *with System-no Stress*, and *without System-no Stress*. As dependent variables, we measured the perceived and observed SA in all experimental conditions, as well as the workload and overall UX in the conditions with the System. To this end, the SART questionnaire [136] was administered for estimating the perceived SA, whereas the SAGAT query technique [50] was employed for the observed SA. For measuring the workload and overall UX, the NASA-TLX [75] and UMUX-Lite [102] questionnaires were completed. All questionnaires employed in this study are standardized, ensuring the validity and accuracy of results.

5.2.2 SAGAT queries

To acquire an objective measure of the participants' SA in the different experimental conditions, the SAGAT query technique was employed, an online probe method based on queries during arbitrary freezes in a simulation [50]. This method has been shown to have a high degree of validity and reliability, and is one of the best publicized and most widely utilized measure of SA, along with the SART questionnaire [48]. The SAGAT questions for our application domain were developed in line with the SA requirements highlighted in the requirements elicitation and analysis procedure (see Annex C), and were evaluated by 2 LEA experts. They were administered at arbitrary time points, appearing in the participants field of view, during freezes of the simulation videos. They assessed the participants perception of elements in the environment, comprehension of the current situation and

prediction of its future status, corresponding to the three levels of SA depicted in Endsley's Model [45]. In Figure 5.3, an example of a SAGAT query is depicted for the video in Figure 5.1.

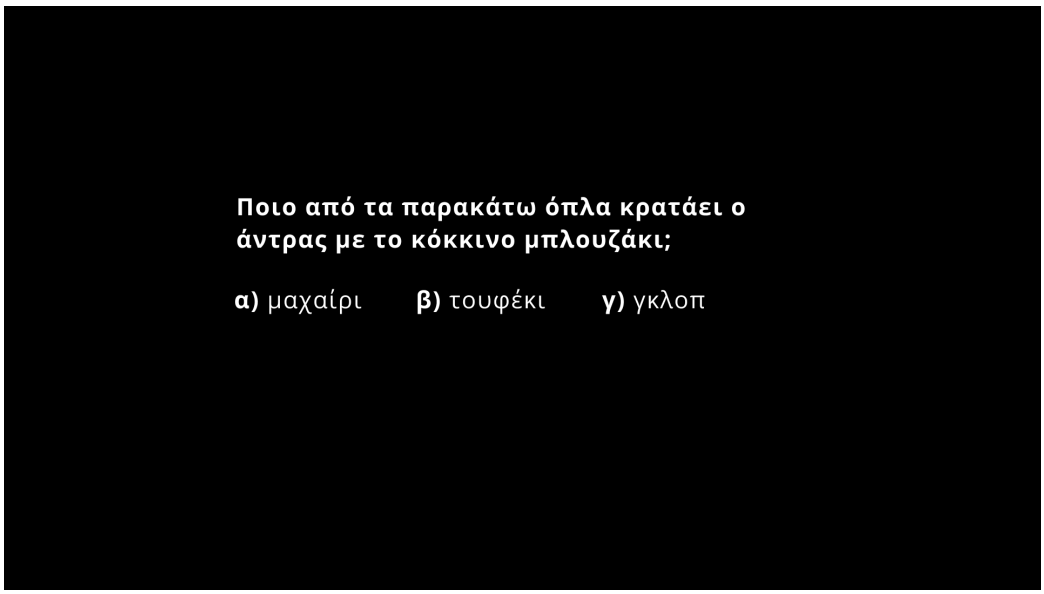

Figure 5.3: One of the SAGAT queries, for the simulation video of Figure 5.1

5.2.3 Stress Induction

There exist a diversity of stress induction methods, employing stressors that are either physical, namely environmental and physiological, or psychological/mental, namely cognitive and emotional, or mixed [19]. In our experiment, we were interested in experimentally inducing psychological stress, an integral part of LEAs working conditions. Mental Arithmetic (MA) tests are a reliable mental stress induction technique, utilized in many studies [65]. In our experiment, we employed the Paced Auditory Serial Addition Test (PASAT) [147], which is a neuropsychological test for assessing attentional processing, also used towards this direction. In particular, we utilized the computerized version provided by the PEBL software [116].

5.2.4 Apparatus and Videos

To simulate the experience of policing using AR glasses, the VR Google Cardboard Headset was utilized. The simulation videos were streamed on a Google Pixel 5 Android Phone, placed inside the headset. They comprised of 8 short ($<$ 1 min) videos, portraying a diversity of staged terrorist attacks in different situations and contexts. In order to avoid detection errors, which could interfere with the evaluation of our approach, the videos were manually annotated. Each detection

was associated with appropriate information, depending on the simulated scenario (e.g. the carried weapons of a foe), which was stored in the Knowledge base and provided to the DM at run-time. The decisions of the DM were propagated to the Visualizer, which displayed the video augmented with appropriate GUI elements. In Figure 5.4, frames from different simulation videos are displayed, augmented with GUI elements.

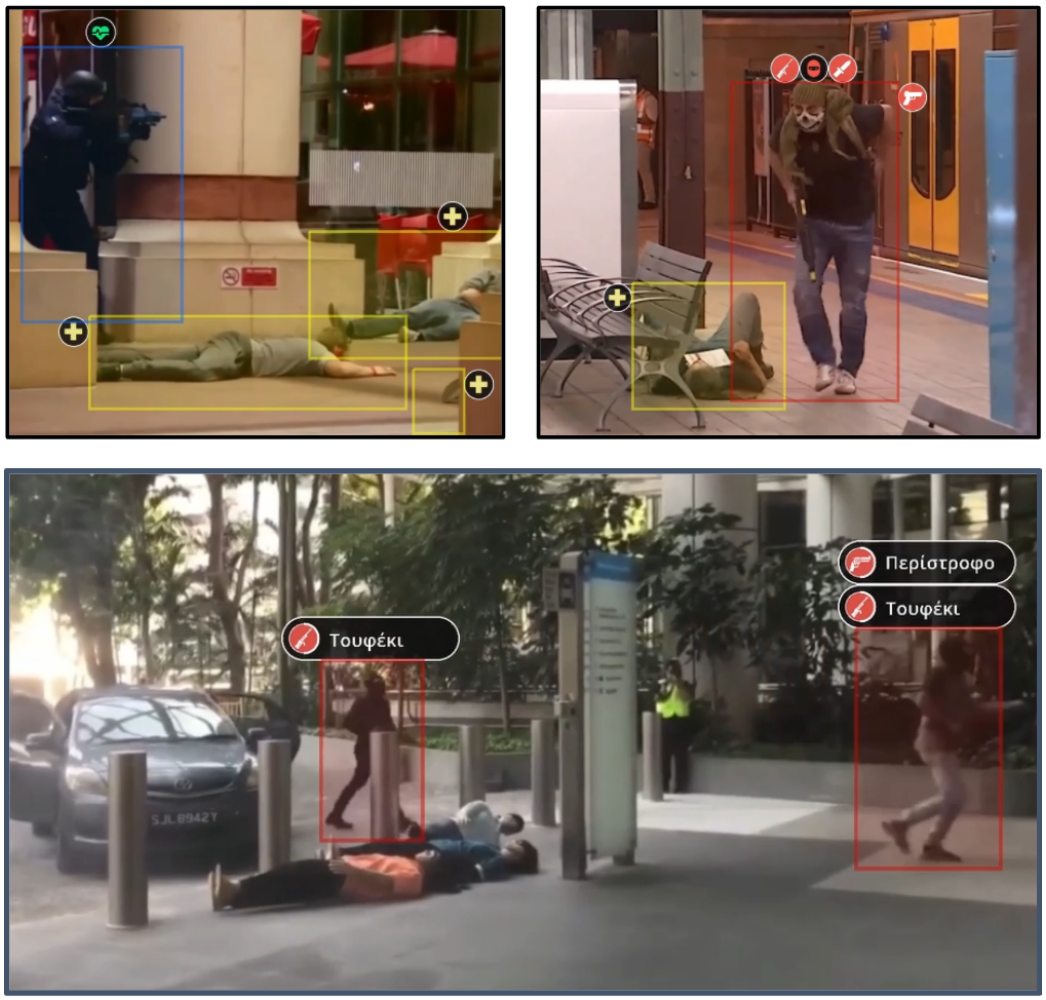

Figure 5.4: Collage of frames from different simulation videos

5.3 Procedure

The experiment was structured in three phases: introduction, main part of the study, and debriefing (see Annex B). Participants were first welcomed and explained the aim and objectives of the study. After signing a consent form, they completed a demographic questionnaire, which included questions regarding their

age range and gender, as well as inquires regarding their professional experience with different policing tasks and incidents (e.g. terrorist attacks and bomb defusals). Then, in order to familiarize them with the system, they were introduced to the GUI Component Types and their LoDs, during a short 5min presentation. Finally, they were asked to calibrate the AR HMD to ensure that the were comfortable with viewing content and to perform a short example in order to familiarize themselves with the task, going through a short video and being asked a few exemplary questions.

In the main part of the experiment, each experimental condition was preceded by a stress manipulation task. In the case of a forthcoming *Stress* condition, the participants performed the PASAT test for 5 minutes, otherwise they watched a relaxing video for 5 minutes, to prepare for the *no Stress* condition. Then, they watched 2 simulation videos with or without the system, depending on the condition, during which they answered to the corresponding SAGAT queries. Finally, the condition concluded by completing the respective questionnaires, which included the SART questionnaire, followed by the NASA-TLX and UMUX-Lite questionnaires in the *with System* conditions. This process continued for all 4 experimental conditions, with different simulation videos in each one. At the end of the experiment, the participants were debriefed. The full study lasted for approximately 60 mins per user.

5.4 Participants

We recruited 20 participants, 3 female and 17 male, between 18 and 54 years, from different Law Enforcement Agencies in Greece. All but 3 participants had no prior experience with AR systems and one did not wish to indicate. Most of the participants (80%) were experienced LEAs with more than 10 years of professional experience. Furthermore 85% had at least some experience with crime arrests or terrorism, with 60% having more than 5 years of expertise. On the other hand, 80% had no experience with diffusion of explosives, while 60% were inexperienced in crisis management or healthcare provision, and hostage situations. Figures 5.5, 5.6 and 5.7, show the distribution of participants age, professional expertise and experience in policing tasks, respectively.

5.5 Results

In this section, the results of the experiment with respect to our initial hypotheses are presented. We first demonstrate the results with respect to the SA of the users, in the stress and non-stress condition. We present SART and SAGAT results with Conclusions in each condition and discussing the qualitative feedback received. We then show the results regarding the workload of the participants while using the system, under stress and non-stress conditions, and compare them to findings from

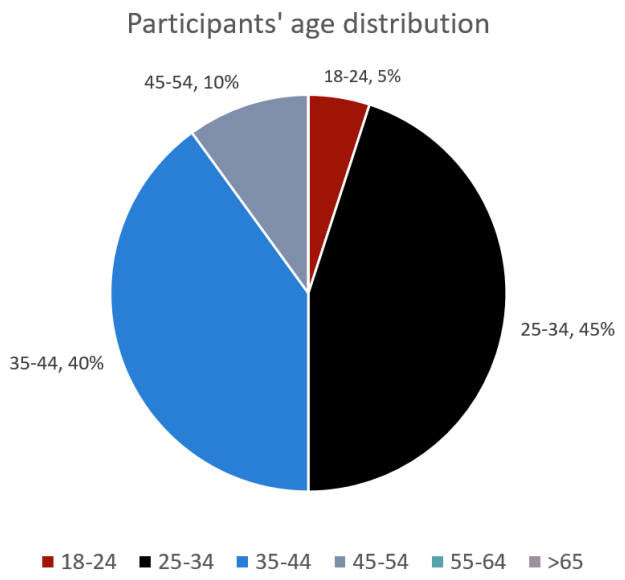

Figure 5.5: Participants' age distribution

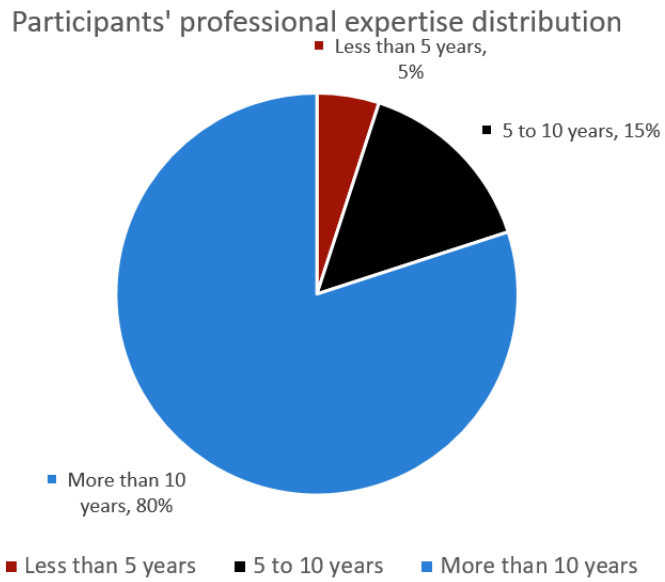

Figure 5.6: Participants' professional expertise distribution

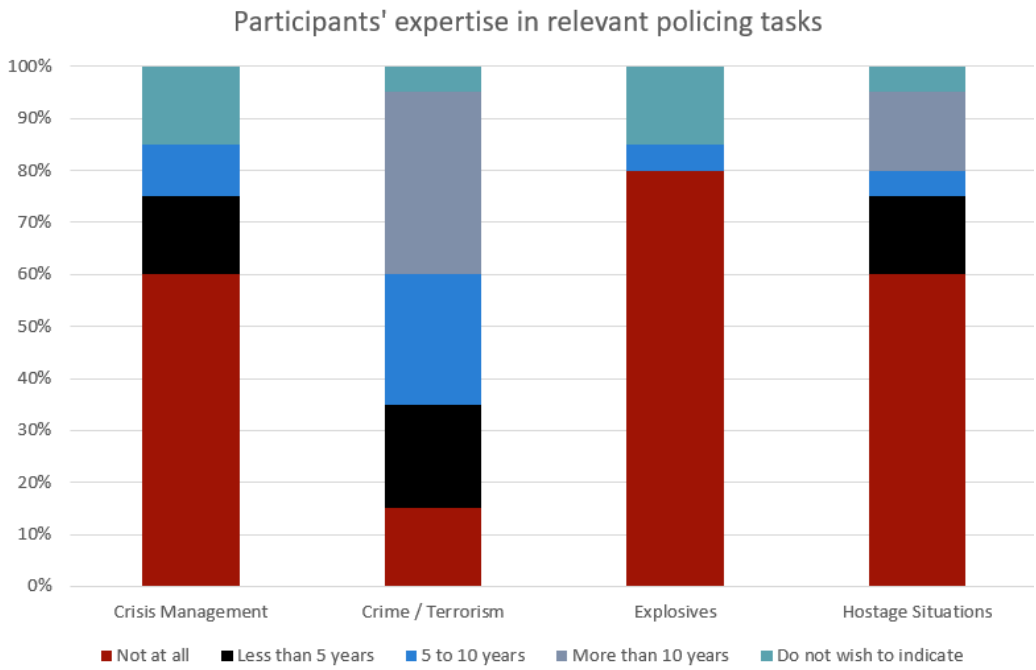

Figure 5.7: Participants' expertise in relevant policing tasks

a study with police officers in a field shooting exercise. Subsequently, User Experience results are presented, including the results from the UMUX-Lite questionnaire and qualitative feedback from the participants. We continue by examining the impact of stress and of professional expertise, as well as the observed SA, for the 3 levels of SA. Finally, the results of correlation analysis of all the measurements acquired in this study are presented.

5.5.1 Situational Awareness

In order to study the Situational Awareness (SA) of the participants, under stress and non-stress conditions (*H1a* and *H1b*), the results from the SART questionnaire and the SAGAT query technique were analysed, assessing the perceived and observed SA respectively. In the SART questionnaire, participants rate their own perception regarding their SA with respect to ten dimensions after the simulation is completed. These ten dimensions are classified into three main subscales: *Attentional Demand*, *Attentional Supply* and *Understanding*. The score for each subscale is calculated as the sum of the participant's rating in each of the subscale's questions. The final SART score is calculated as per equation 1 below.

$$SART = Understanding - (AttentionalDemand - AttentionalSupply) \quad (1)$$

We note that data from one participant were eliminated from the dataset, since it was incomplete. For scoring the results of the SAGAT query technique, each correct

response to a question acquired one score point, whereas erroneous responses did not receive any points. Then, all the individual scores for each participant were accumulated and divided by the total number of questions the participant was asked, acquiring the final SAGAT score, which represents the percentage of their correct responses.

Overall, use of the system improved perceived and observed SA, by 25.63% and 9.25% respectively, cumulatively for both stress states. In particular, in the case of stressful conditions, perceived SA was improved by 30%, whereas observed SA by 3.95%. In non-stressful conditions, perceived SA was improved by 15.65%, while observed SA by 15%. These results are displayed graphically in Figure 5.8 for the perceived SA and in Figure 5.9 for the observed SA, with the error bars indicating the 95% confidence interval (CI). In the following sections, we analyse the results of the perceived and observed SA, for each stress condition, namely the *Stress Condition* and the *Non-stress Condition*.

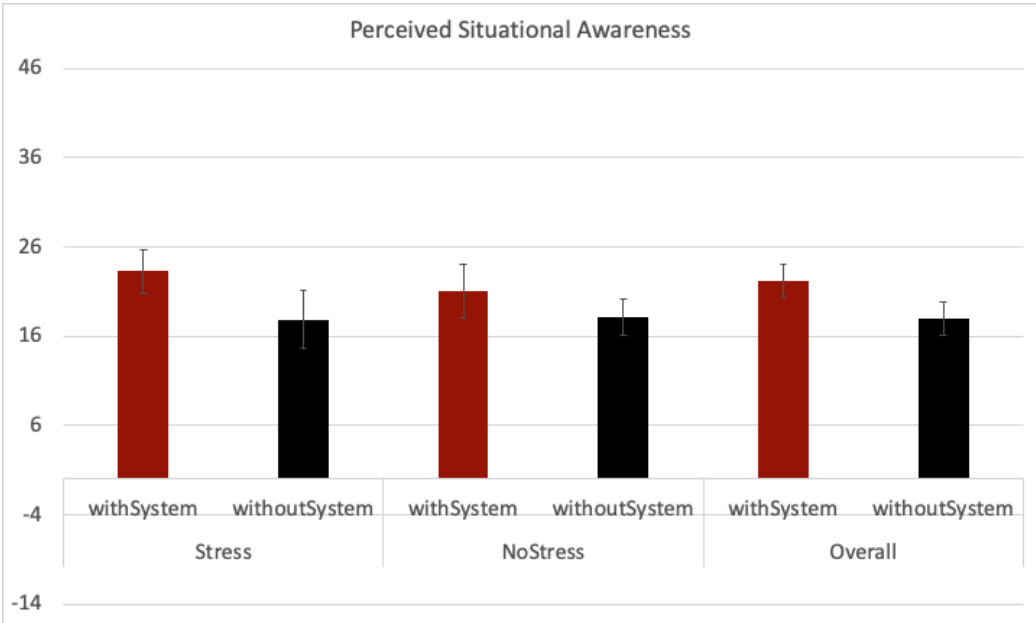

Figure 5.8: Observed Situational Awareness (SA), in both stress states and overall

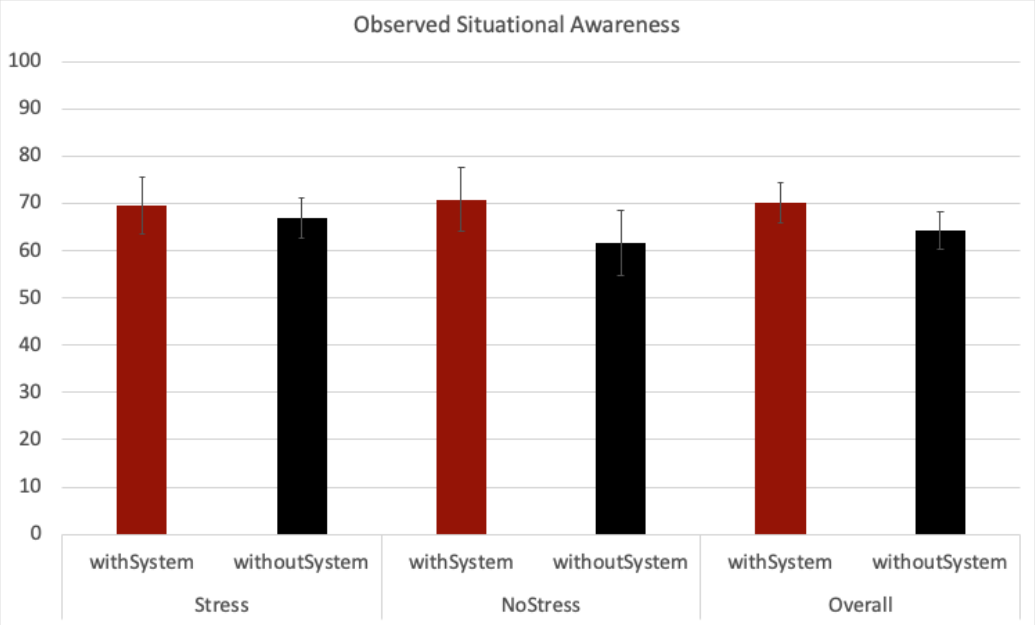

Figure 5.9: Perceived Situational Awareness (SA), in both stress states and overall

5.5.1.1 Stress condition

In the following paragraphs, we analyse the SA of the participants, with and without the system, while they were under stress (*hypothesis H1a*).

Perceived SA The overall perceived Situational Awareness (SA) was enhanced when using the system in the *Stress Condition*, as shown in Figure 5.10, and when comparing Table 5.1 with Table 5.2.

Furthermore, taking into account that SART scores are limited to values between -14 and 46, it turns out that when using the system, all participants' SART scores were above the midpoint of this range (16), as can be observed in Figure 5.10 and Table 5.1, which reflects that they had good SA during the simulation scenarios, corresponding to this condition. The blue line in Figure 5.10 indicates this midpoint, whereas the red and black lines indicate the average perceived SA of all participants, with and without use of the system respectively. On the contrary, when not using the system, 30% of the participants had a SART score below the midpoint of this range, as shown in Figure 5.10 and Table 5.2, highlighting that some of them perhaps felt that their SA was not so good. In Figure 5.11, Figure 5.12 and Figure 5.13, we can see the SART scores of the participants for each subscale, namely *Attentional Demand*, *Attentional Supply* and *Understanding*, respectively. In more detail, as shown in Table 5.2 attentional demand was perceived higher when not using the system, whereas attentional supply and understanding were perceived higher with the system.

To compare the results of the overall SA, as well as all the individual SART subscales, when using the system and without it, paired two-tailed t-tests were conducted. Statistically important differences were identified for the *Understanding* subscale, when using the system (M=15.05, SD=2.61) and without it (M=12.42, SD=3.25); t(18)=2.78. p=0.02. Furthermore, statistically important differences were identified for the overall SA score between the two conditions of using the system (M=23.26. SD=5.08) and without it (M=17.89. SD=6.87); t(18)=2.44, p=0.025.

Table 5.1: SART SA results for the Stress condition With the System

With the System				
	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
Mean	13.32	21.53	15.05	23.26
Min	5	16	10	16
Max	21	28	19	38
Range	16	12	9	22
SD	5.28	3.03	2.61	5.08
95%CI	[10.77, 15.86]	[20, 22.98]	[13.79, 16.31]	[20.82, 25.71]

Table 5.2: SART SA results for the Stress condition Without the System

Without the System				
	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
Mean	15.26	20.74	12.42	17.89
Min	8	9	7	-1
Max	21	28	19	29
Range	13	19	12	30
SD	4.01	4.15	3.25	6.87
95%CI	[13.33, 17.2]	[18.74, 22.74]	[10.85, 13.99]	[14.58, 21.21]

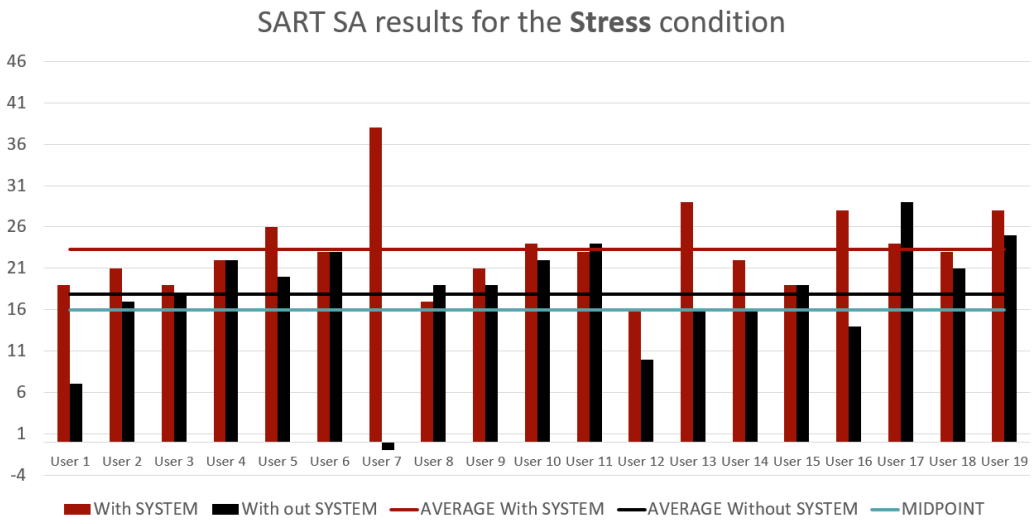

Figure 5.10: SART SA results for the Stress condition

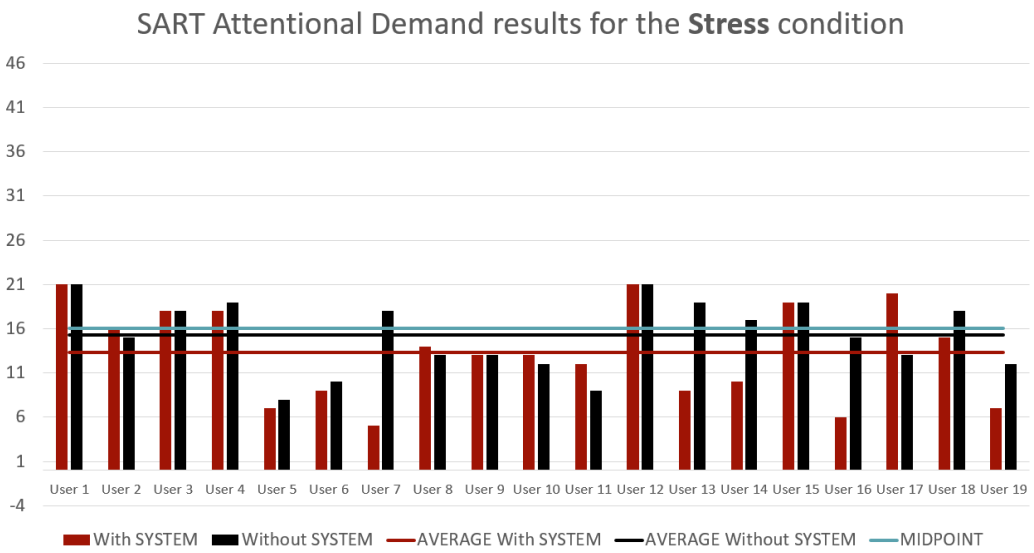

Figure 5.11: SART Attentional Demand results for the Stress condition

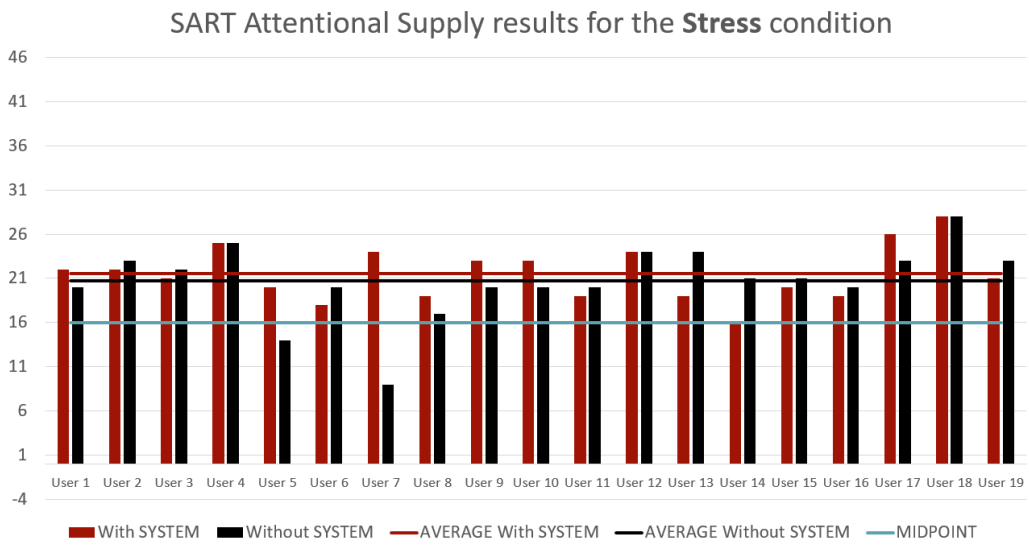

Figure 5.12: SART Attentional Supply results for the Stress condition

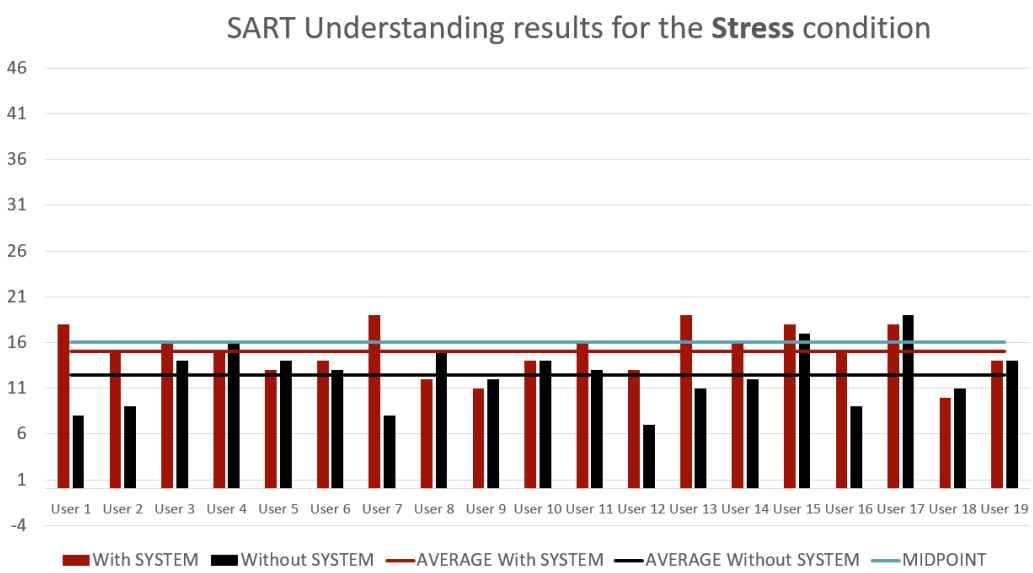

Figure 5.13: SART Understanding results for the Stress condition

Observed SA On average, users of the system achieved a better SA score (M=69.52, SD=12.66) than participants without the system (M=66.88, SD=8.96), as can be observed in Table 5.3 and Figure 5.14.

At the same time, system users exhibited higher variance in their scores, implying that the system did not have the same positive impact on all participants' scores. Statistical analysis through paired two-tailed t-tests comparing the SAGAT

scores when using the system and without it did not reveal any statistically important differences between the two conditions. A potential reason for this is that the information displayed by the system is less detailed in the case that LEAs are stressed. In particular, the system, as a general approach, adapts the UI based on the context, and in stressful conditions prefers to decrease the LoD of Components, communicating information when possible through icons. In this context, it could be the case that the short training that preceded the evaluation was not adequate to familiarize participants with all the different icons and their meanings. This was also pointed out by a considerable number of participants (35%) during the debriefing session, saying that they would require additional training prior to actually using it and stating that they felt that they got better with the system the more they used it.

Table 5.3: SAGAT results for the Stress condition With the System and Without the System

	With the System	**Without the System**
Mean	69.52	66.88
Min	42.86	46.67
Max	93.33	80
Range	50.48	33.33
SD	12.66	8.96
95%CI	[63.6, 75.44]	[62.69, 71.07]

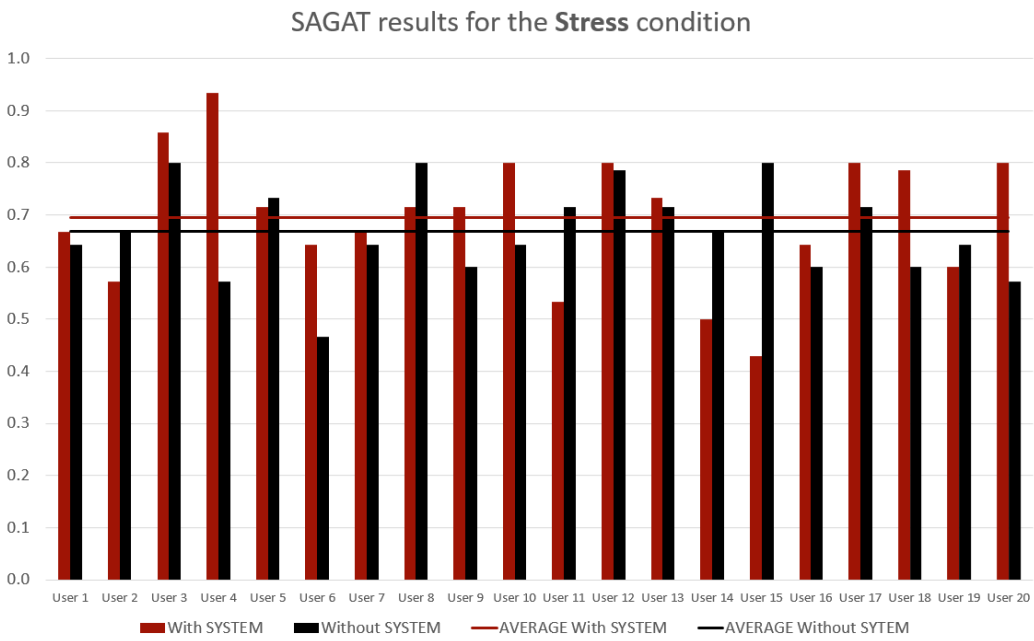

Figure 5.14: SAGAT results for the Stress condition

Conclusions In conclusion, participants perceived their SA to be enhanced when using the system as opposed to not using it, as indicated by the results of the SART questionnaire. Moreover, their observed SA also improved, when measured through the SAGAT query technique. This confirms our hypothesis that the system enhances the SA, in stress conditions (*H1a*). It should be noted that with respect to the perceived SA, differences were statistically important, whereas for observed SA they were not.

5.5.1.2 Non-stress condition

In the following paragraphs, we analyse the SA of the participants, with and without the system, in the non-stress condition (*hypothesis H1b*).

Perceived SA On average, participants perceived that they achieved higher SA with the system, also in the *Non-stress Condition*, as can be observed in Figure 5.15, and when comparing Table 5.4 with Table 5.5. Moreover, when using the system, 20% of participants achieved a SART score lower than the midpoint of the SART scores range. This percentage doubled to 40% when participants were not using the system.

In Figure 5.15, we can see the SART SA scores of each individual participant, with and without the system, whereas in Figure 5.16, Figure 5.17 and Figure 5.18, we can see the SART scores of the participants for each subscale, namely *Attentional Demand*, *Attentional Supply* and *Understanding*, respectively. At the level of

individual subscales, they perceived that greater *Attentional Demand* was required without the system and slightly less *Attentional Supply*, while *Understanding* was, on average, perceived as enhanced when using the system.

Statistical analysis through paired two-tailed t-tests comparing the results of overall SA and all the individual SART subscales when using the system and without it, did not reveal any statistically important differences for any of the involved scales.

Table 5.4: SART SA results for the Non-Stress condition With the System

With the System				
	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
Mean	13.47	20.42	14.05	21
Min	4	13	11	7
Max	21	28	19	32
Range	17	15	8	25
SD	5.35	4.4	2.76	6.19
95%CI	[10.9, 16.05]	[18.3, 22.54]	[12.72, 15.38]	[18.02, 23.98]

Table 5.5: SART SA results for the Non-Stress condition Without the System

Without the System				
	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
Mean	14.53	20.11	12.58	18.16
Min	6	13	9	9
Max	20	27	19	28
Range	14	14	10	19
SD	4.6	3.3	2.73	4.14
95%CI	[12.31, 16.74]	[18.52, 21.69]	[11.26, 13.9]	[16.16, 20.15]

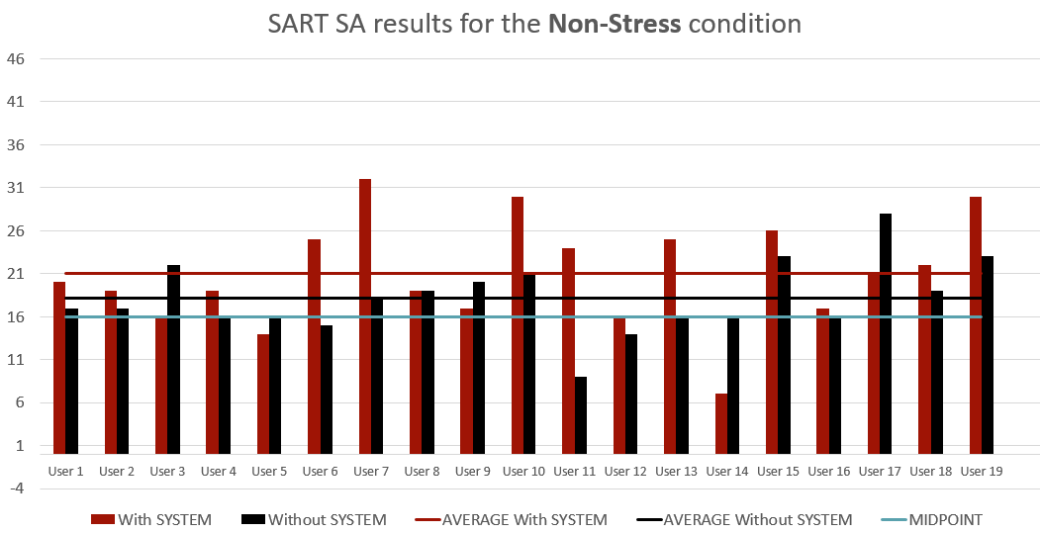

Figure 5.15: SART SA results for the Non-Stress condition

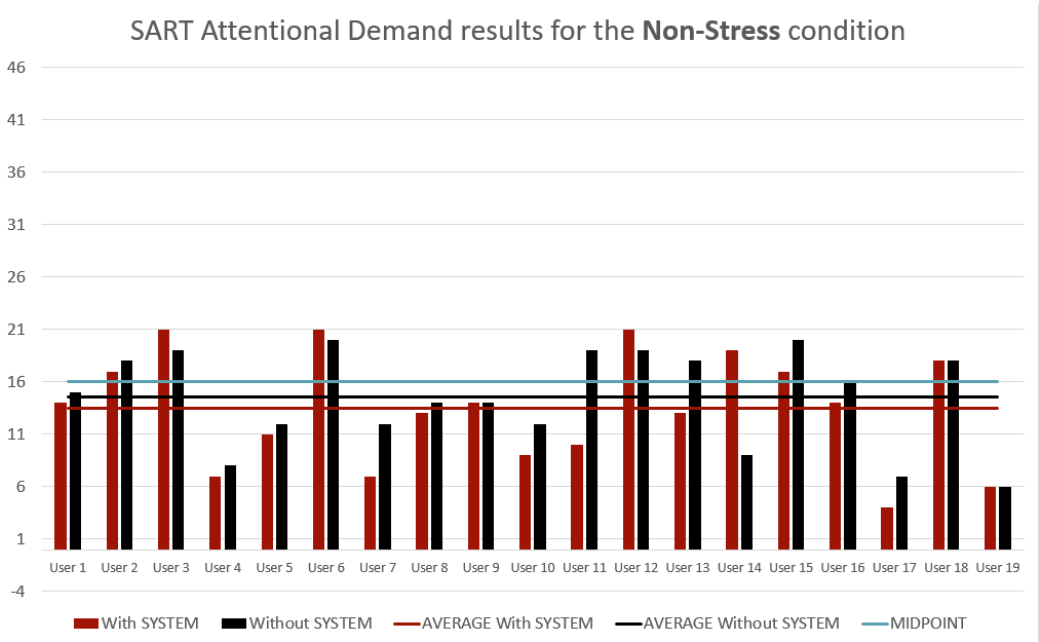

Figure 5.16: SART Attentional Demand results for the Non-Stress condition

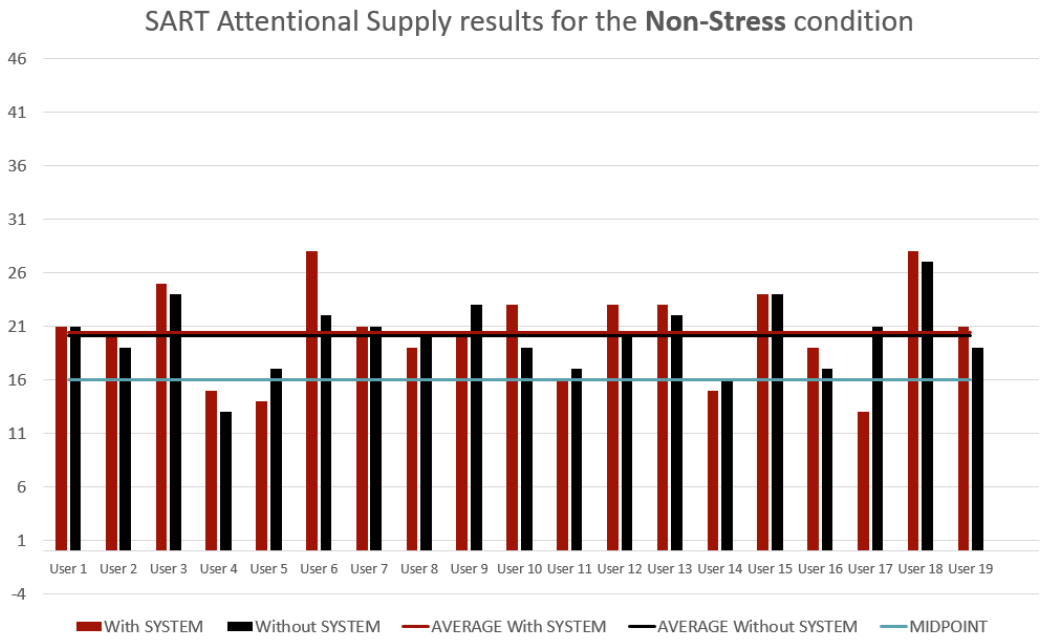

Figure 5.17: SART Attentional Supply results for the Non-Stress condition

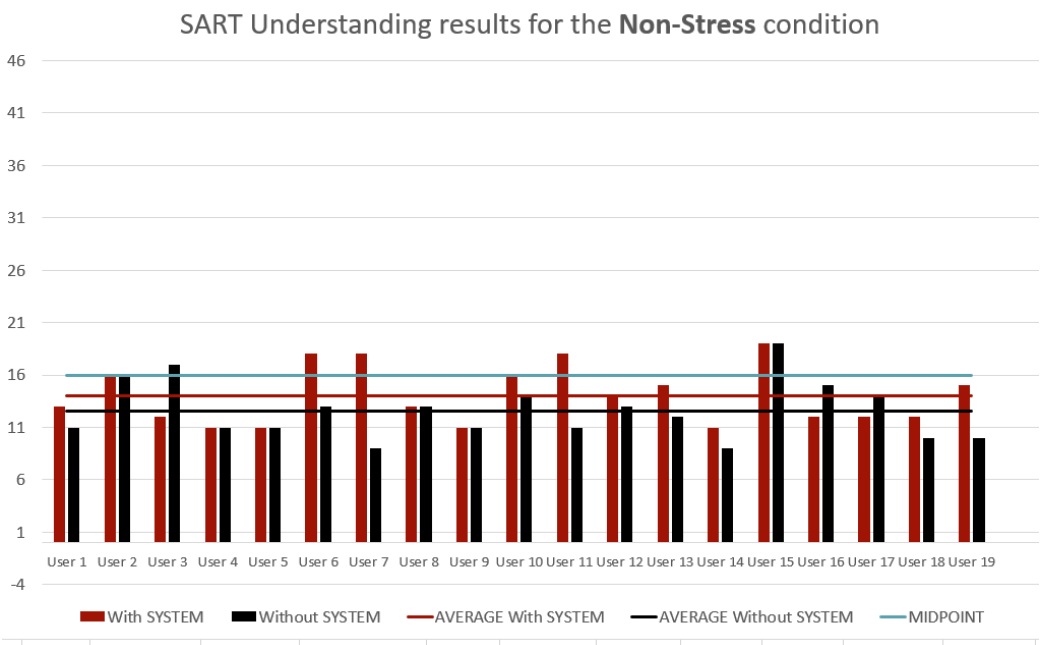

Figure 5.18: SART Understanding results for the Non-Stress condition

Observed SA In the *Non-stress Condition*, it is evident from the results in Table 5.6 and Figure 5.19, that the participants' observed SA when using the system outperformed their observed SA without it. This conclusion is further confirmed through a paired two-tailed t-test that compared the SAGAT score results in the two cases, yielding statistically important differences when using the system (M=70.86, SD=14.62) and without it (M=61.62, SD=14.49); t(19)=2.24, p=0.03.

Table 5.6: SAGAT results for the Non-Stress condition With the System and Without the System

	With the System	**Without the System**
Mean	70.86	61.62
Min	40	28.57
Max	93.33	86.67
Range	53.33	58.1
SD	14.62	14.49
95%CI	[64.01, 77.7]	[54.84, 68.4]

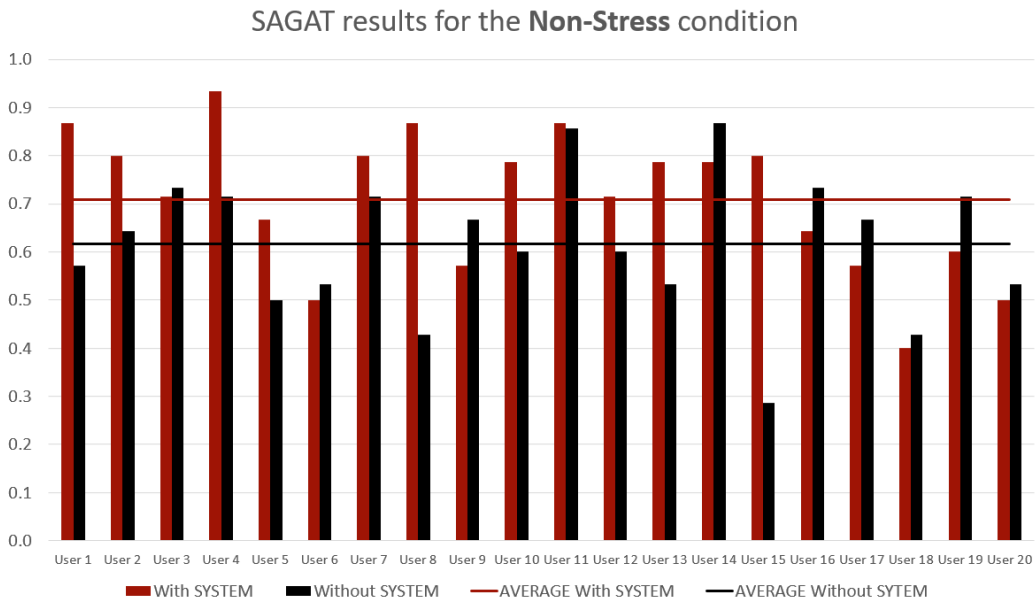

Figure 5.19: SAGAT results for the Non-Stress condition

Conclusions In conclusion, participants' performance in terms of SA improved when using the system as opposed to not using it, as measured through the SAGAT

query technique. At the same time, they also perceived an enhanced SA, indicated by the results of the SART questionnaire. Furthermore, their observed SA exhibited statistically important differences with the system, in comparison to without it. As a result, it can be concluded that *H1b* is supported.

5.5.1.3 Qualitative feedback

As analysed so far in this section, the system enhances SA when using the system, both in stress and non-stress conditions, confirming our hypotheses (*H1a* and *H1b*). This finding is also supported by qualitative feedback, solicited during the debriefing session. Overall, the participants' reaction to the system, in terms of their SA (as their general feeling) was positive. In particular, 12 participants (60% of the total sample) were strongly positive about the system's usefulness in improving their SA, providing statements like "it certainly will", "definitely", etc. Moreover, 3 participants (15% of the total sample) were rather positive, stating, for instance, that the system would be helpful, but it requires training, or that it is generally useful but not always. Another 15% of the participants were neutral, highlighting that the system may be useful in some circumstances, whereas in others it might not be. Finally, one participant (5% of the total sample) was rather negative, suggesting that they would not normally use it, unless they were facing a crisis that has escalated, and another participant (5% of the total sample) was strongly negative, saying that they would prefer to not use the system.

5.5.2 Workload

In order to study the workload of the participants while using the system, under stress and non-stress conditions (hypotheses *H2a* and *H2b*), the results from the NASA task load index (NASA-TLX) questionnaire were analysed. The NASA-TLX is a standardized questionnaire for assessing the perceived workload by participants, across 6 dimensions, to determine an overall workload rating. The overall score, as well as the score of each dimension is in the range from 0 to 100. These dimensions are the following:

- Mental demand: how much thinking, deciding, or calculating was required to perform the task.
- Physical demand: the amount and intensity of physical activity required to complete the task.
- Temporal demand: the amount of time pressure involved in completing the task.
- Effort: how hard does the participant have to work to maintain their level of performance?
- Performance: the level of success in completing the task.

- Frustration level: how insecure, discouraged, or secure or content the participant felt.

For the calculation of the overall workload score for a participant, the score of the *Performance* dimension is inverted, and the average across all dimensions is computed.

5.5.2.1 Stress condition

In this paragraph, we analyse the workload of the participants in stress conditions, while using the system (hypothesis *H2a*). The average of the overall workload score was **58.33**, while the score for each of the NASA-TLX dimensions is summarized in Table 5.7. In Figure 5.20, the workload is depicted for each dimension of each participant (bars), along with the overall workload score for each participant (line). From the above, we can observe that, on average, the mental workload (65.79) was rather higher than the midpoint (50), while the perceived effort (58.68) and temporal workload (53.16) were also higher. At the same time, it is notable that, on average, the performance score was also highly rated (62.89), whereas the frustration (35.53) and physical workload (20) was very low; the latter was anticipated given the simulated nature of the study.

Table 5.7: Perceived workload using the System in Stress condition

	Mental	**Physical**	**Temporal**	**Performance**	**Effort**	**Frustration**
Mean	65.79	20	53.16	62.89	58.68	35.53
Min	15	0	5	35	20	0
Max	100	100	100	90	90	80
Range	85	100	95	55	70	80
SD	25.89	27.69	28.97	17.74	17.7	23.86
95%CI	[53.31, 78.27]	[6.65, 33.35]	[39.19, 67.12]	[54.34, 71.45]	[50.15, 67.22]	[24.03, 47.02]

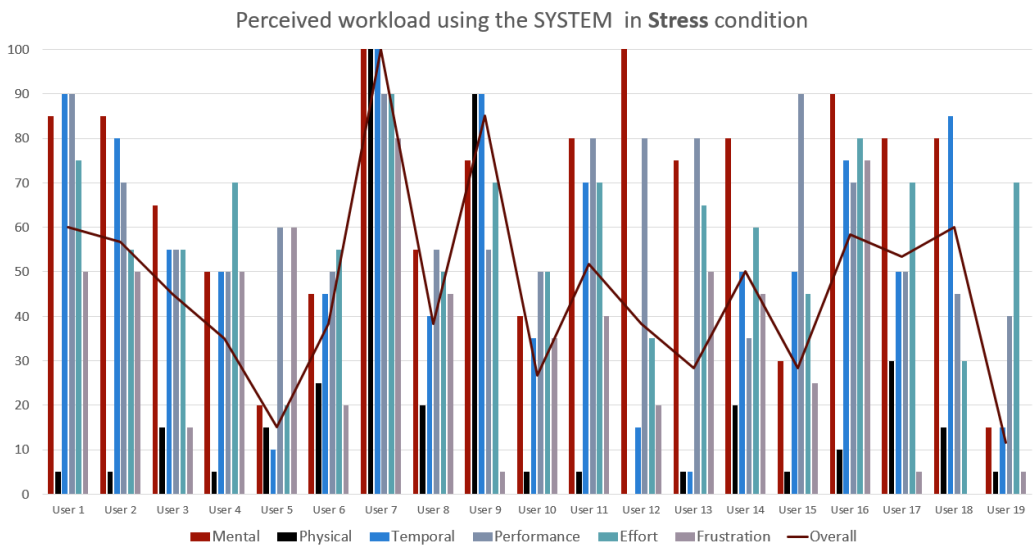

Figure 5.20: Perceived workload using the System in Stress condition

5.5.2.2 Non-stress condition

In this paragraph, we analyse the workload of the participants in non-stress conditions, while using the system (hypothesis *H2b*). The average of the overall workload score was **43.54**, while the score for each of the NASA-TLX dimensions is summarized in Table 5.8. In Figure 5.21, the workload is depicted for each dimension of each participant. From the above, we can observe that, the overall workload score (43.54) is below the midpoint (50) and considerably lower than in stressful conditions (58.33), indicating that when stressed, the workload of LEAs is significantly higher, a finding which we intuitively expected. Moreover, on average, similar to stressful conditions, the mental workload (64), the temporal workload (54.5) and the performance scores (60.5) were higher than the midpoint (50), whereas the frustration (35.5) and physical workload (22.5) were considerably lower than the midpoint. However, when the participants were not stressed while using the system, the average effort (47.25) was lower than when they were stressed (58.68). This could be explained by the mental state of the user, as well as the fact that the system during stressful conditions decreases the LoD, displaying the information through icons without any textual information, thus potentially requiring more effort to perceive it, as indicated by the debriefing, detailed in a following section.

Table 5.8: Perceived workload using the System in Non-Stress condition

	Mental	**Physical**	**Temporal**	**Perform-ance**	**Effort**	**Frustration**
Mean	64	22.5	54.5	60.5	47.25	33.5
Min	10	0	5	20	10	5
Max	100	100	100	95	90	85
Range	90	100	95	75	80	80
SD	28.45	29.13	31.12	20.45	24.14	26.01
95%CI	[50.68, 77.32]	[8.87, 36.13]	[39.94, 69.06]	[50.93, 70.07]	[35.95, 58.55]	[21.33, 45.67]

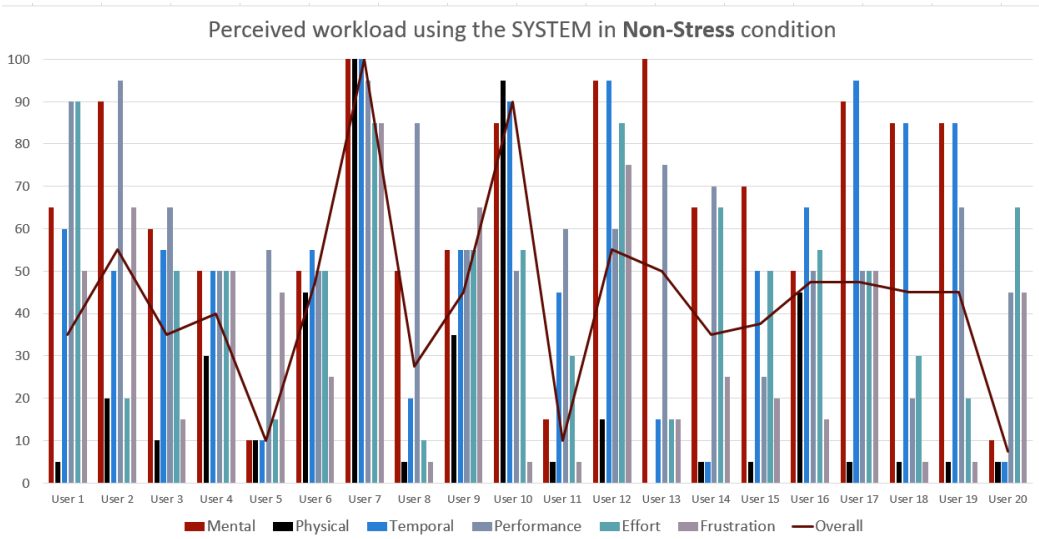

Figure 5.21: Perceived workload using the System in Non-Stress condition

5.5.2.3 Comparison with field shooting exercise

In order to better understand these scores, Table 5.9 and Figure 5.22 summarize these results in comparison to findings from a study with police officers in a field shooting exercise [121]. It is evident that the perceived workload, when using the system for policing tasks, is in general aligned with findings from actual policing tasks. This holds true for both stress and no-stress conditions. Thus, the hypotheses *H2a H2b* that the system does not impose workload are supported. The main difference among the 6 dimensions was that the physical workload was found considerably lower in our study. This was an expected finding, since this study was a simulation and did not require actual physical effort. However, despite the

simulated nature of this study, this dimension was meaningful in order to detect potential physical workload induced by the AR HUD. In addition to this, an explicit question regarding nausea was addressed to participants during the debriefing session, which yielded negative results for most of the participants (85%). Moreover, the Participants who experienced nausea rated it as of moderate impact (6 out of 10).

Table 5.9: Summary of results in comparison to findings from a study with police officers in a field shooting exercise

	System stress condition	**System no stress condition**	**Warmup policing task**	**Flashlight policing task**	**Barrel policing task**	**Metal policing task**
Mental	65.79	64	52	59	68	65
(M, SD)	25.89	28.45	26	23	21	24
Physical	20	22.5	28	40	62	53
(M, SD)	27.69	29.13	23	26	24	25
Temporal	53.16	54.5	38	45	67	62
(M, SD)	28.97	31.12	26	26	22	25
Performance	62.89	60.5	63	61	55	52
(M, SD)	17.74	20.45	24	23	23	22
Effort	58.68	47.25	48	53	65	62
(M, SD)	17.7	24.14	23	24	20	20
Frustration	35.53	33.5	29	38	47	49
(M, SD)	23.86	26.01	23	23	27	25

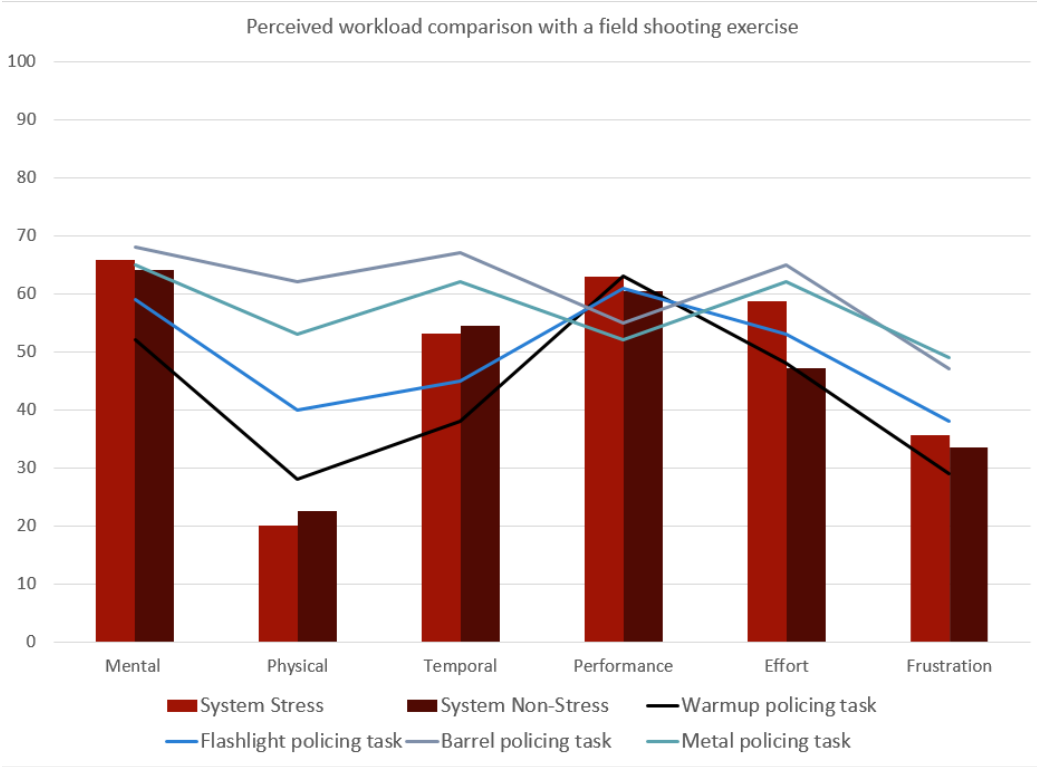

Figure 5.22: Perceived workload comparison with a field shooting exercise

5.5.3 User Experience

In order to study the User Experience (UX) of the participants while using the system, under stress and non-stress conditions (hypotheses *H3a* and *H3b*), the results from the UMUX-Lite questionnaire were analysed. The UMUX-Lite is a standardized questionnaire that measures the perceived ease-of-use and the perceived usefulness regarding a system, in a scale from 1 to 7.

5.5.3.1 UMUX-Lite Results

In the stress condition, the overall UX score across participants was 5.03 out of 7 (stdev: 1.50), with 95%CIs [4.31, 5.75]. The question regarding whether the system meets their requirements (usefulness) had an average score of 4.95 (stdev: 1.35), with 95%CIs [4.30, 5.60]. Overall, 31.57% of participants generally agreed that it meets their requirements (voted 6 or 7). At the same time, the scores also indicate that the system is easy to use (usable), with an average score of 5.11 (stdev: 1.94) and 95%CIs [4.17, 6.04]. Overall, 52.63% of participants generally agreed that it is easy to use (voted 6 or 7). These results can be observed in Table 5.10 and Figure 5.23. The scores of each participant are displayed in Figure 5.24. The blue columns indicate the usefulness, the purple columns the usability and the yellow

line the average of the two.

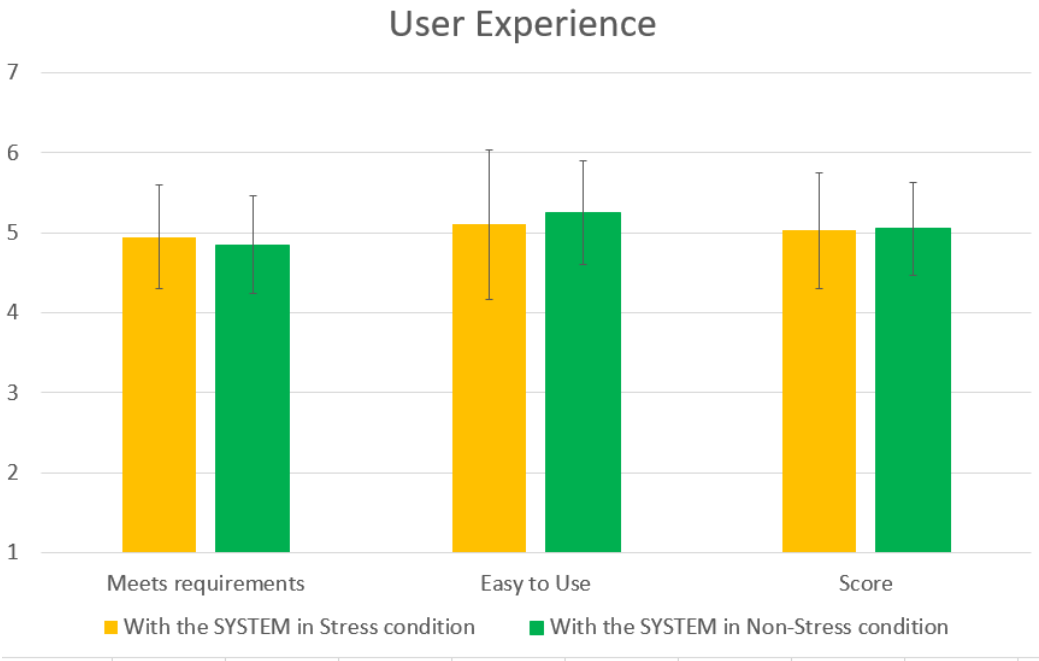

Figure 5.23: User Experience

Table 5.10: User Experience in Stress condition

	Meets requirements	**Easy to use**	**Overall Score**
Mean	4.95	5.11	5.03
Min	2	1	2.5
Max	7	7	7
Range	5	6	4.5
SD	1.35	1.94	1.5
95%CI	[4.3, 5.6]	[4.17, 6.04]	[4.31, 5.75]

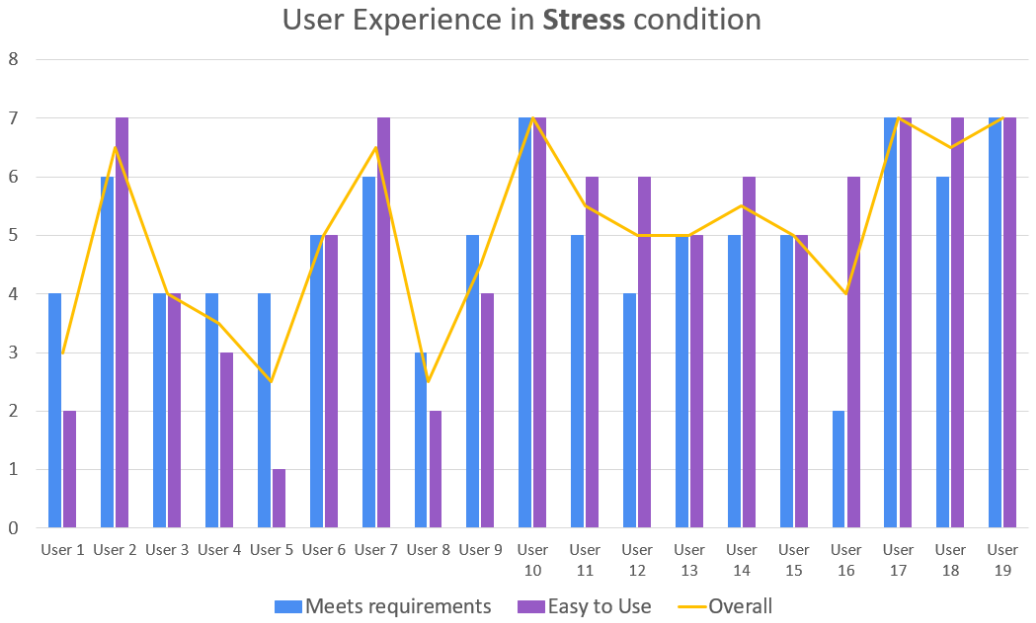

Figure 5.24: User Experience in Stress condition

Furthermore, as summarized in Table 5.11 and Figure 5.23, in the non-stress condition, the overall UX score across participants was 5.05 out of 7 (stdev: 1.23), with 95%CIs [4.4, 5.63]. The question regarding whether the system meets their requirements (usefulness) had an average score of 4.85 (stdev: 1.31), with 95%CIs [4.24, 5.46]. Overall, 30% of participants generally agreed that it meets their requirements (voted 6 or 7). The results also indicate that the system is easy to use (usable), with an average score of 5.25 (stdev: 1.37), with 95%CIs [4.61, 5.89]. Overall, 45% of participants generally agreed that it is easy to use (voted 6 or 7). The scores of each participant are displayed in Figure 5.25.

Table 5.11: User Experience in Non-Stress condition

	Meets requirements	**Easy to use**	**Overall Score**
Mean	4.85	5.25	5.05
Min	2	2	2
Max	7	7	7
Range	5	5	5
SD	1.31	1.37	1.23
95%CI	[4.24, 5.46]	[4.61, 5.89]	[4.47, 5.63]

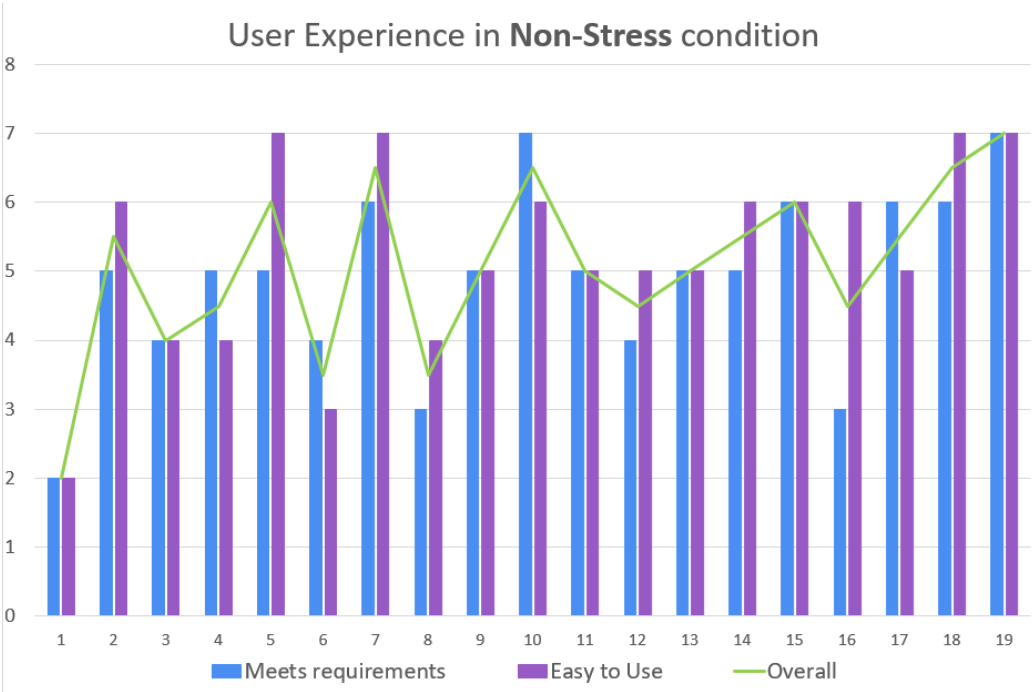

Figure 5.25: User Experience in Non-Stress condition

From the aforementioned results, we can conclude that the hypotheses *H3a* and *H3b* are confirmed, considering that the average overall UX score is above the midpoint of the UMUX-Lite scale (4), both in the stress and non-stress conditions. The same holds for both constituents of the overall UX, namely usefulness and usability for both experimental conditions. Nevertheless, additional insights were sought in the participants' responses to the debriefing session, aiming to identify potential shortcomings with regard to the system's usefulness and usability.

5.5.3.2 Qualitative feedback

During the debriefing session, participants were inquired about how easy it would be to use the system during their daily tasks. It is notable that the majority of the comments received pertained to the device itself (AR HMD) and not the visualized UI. Feedback regarding the UI indicated that training would be required for LEAs, and that, in real-life field operations, the information displayed should not distract them, taking their focus off the actual real-world operation. This remark is aligned with observations from the expert-based evaluation, as well as with requirements identified by police officers during the Co-creation Workshops.

Regarding what participants liked the most about the system, an analysis of their comments highlighted the following aspects (comments are provided as they were given by the participants):

- Increased situation awareness (30% of the participants), by providing an

overview of the field and insights to better understand what is going on

- Information about carried weapons (25% of the participants)
- Identification of foes (20% of the participants)
- Assessment of threats (20% of the participants)
- Stress-related information (15% of the participants)
- Victims' identification and information about their health status (15% of the participants)
- Information richness and usefulness (15% of the participants)
- Clear icons (10% of the participants)

In terms of what they disliked, 20% of the participants commented that, in some cases, too much information was displayed, a valuable remark for future improvements of the system. One participant (5%) identified that they disliked the headset. Although this is a useful remark in terms of highlighting requirements for acceptable headsets for the deployment of the system, it is noted that the headset employed, solely served the simulation needs of this study. Moreover, one participant pointed out that they disliked the detection rectangles in general, and another indicated that they did not like the colour of the victims' highlighting rectangle. Finally, one participant expressed concerns regarding potential attention distraction that might be caused by the system. This is a legitimate concern, since LEAs should focus their attention at the crime scene in front of them. Similar concerns have been raised during the Co-creation Workshops, highlighting users' need for a system that will support their operations in an unobtrusive manner. However, we should note that such a concern was not confirmed by the study; instead, the results indicated that the system assists LEAs in achieving increased SA, without inducing workload.

With respect to additional functionality that was requested, one participant suggested that a Map Component, with LEAs' on the ground positions clearly marked, would be useful. Another participant proposed that the system could provide navigation instructions. It is noted that, both of these features have been implemented as GUI Component Types, but were not included in the current study, in order to avoid overwhelming participants with extraneous information, taking into account that it was their first encounter with the system.

With respect to whether they would eventually use the system, participants' responses were as follows:

- 40% of the participants indicated that they would definitely use it for the benefits it offers

- 30% of the participants would use it under specific preconditions (e.g. by specific members of the team) or in specific circumstances (e.g. when encountering suspicious situations)
- 25% of the participants said that they might use it
- 5% of the participants identified that they would be reluctant to use it

5.5.4 Impact of Stress

Apart from our initial, core research questions, defined in 5.1, we also assessed the impact of stress in the perceived and observed SA, the workload, and UX of the participants, both with and without using the system. Our aim was to identify potential differences in these measures, caused by the UI adaption and the physiological state of the user.

In particular, our hypotheses were defined as follows:

H4. Stress has an impact on all observed measures

H4a. Stress has an impact on perceived SA when using the system.

H4b. Stress has an impact on perceived SA when not using the system.

H4c. Stress has an impact on observed SA when using the system.

H4d. Stress has an impact on observed SA when not using the system.

H4e. Stress has an impact on perceived workload with the system.

H4f. Stress has an impact on perceived UX with the system.

For that purpose, paired two-tailed t-tests were carried out on the participants' following scores:

- *on participants' SART overall scores when using the system, indicating that there is a statistically significant difference between the perceived SA using the system in the stress condition (M=23.26, SD=5.08) and in the non-stress condition (M=21.00, SD=6.19); t(18) = 2.14, p = 0.04. A potential reason for this could be that, in the stress condition, the system minimizes displayed information to avoid overloading the users and allow them to focus on the situation at hand. To this end, it keeps only icons, eliminating textual descriptions. On the contrary, in the no Stress condition, the system keeps textual information along with the icons, thus requiring greater attentional demand from the users. This might explain why the perceived SA score was lower during the stress condition*
- SART Attentional Demand scores when using the system, yielding no significant effect on perceived SA for stress; t(17) = -0.62, p = 0.54

- SART Attentional Supply scores when using the system, yielding no significant effect on perceived SA for stress; t(17) = 1.21, p = 0.24
- SART Understanding scores when using the system, yielding no significant effect on perceived SA for stress; t(17) = 1.77, p = 0.09
- SART overall scores without using the system, yielding no significant effect on perceived SA for stress; t(17) = 0.24, p = 0.81
- SART Attentional Demand scores without using the system, yielding no significant effect on perceived SA for stress; t(17) = 0.31, p = 0.76
- SART Attentional Supply scores without using the system, yielding no significant effect on perceived SA for stress; t(17) = 0.73, p = 0.47
- SART Understanding scores without using the system, yielding no significant effect on perceived SA for stress; t(17) = -0.14, p = 0.89
- SAGAT scores when using the system, yielding no significant effect on observed SA for stress; t(19) = -0.29, p = 0.78
- SAGAT scores without using the system, yielding (again) no significant effect on observed SA for stress; t(19) = 1.29, p = 0.21
- NASA-TLX total score, yielding no significant effect on perceived workload for stress; t(18) = 0.87, p = 0.39
- NASA-TLX mental workload score, yielding no significant effect on perceived mental workload for stress; t(18) = 0.41, p = 0.68
- NASA-TLX physical workload score, yielding no significant effect on perceived physical workload for stress; t(18) = -1, p = 0.33
- NASA-TLX temporal workload score, yielding no significant effect on perceived temporal workload for stress; t(18) = -0.86, p = 0.4
- NASA-TLX performance achieved score, yielding no significant effect on perceived performance achieved for stress; t(18) = 1, p = 0.32
- *NASA-TLX effort score, indicating that there is a statistically significant difference on the perceived effort between using the system in stress conditions (M=58.68, SD=17.7) and in non-stress conditions (M=47.25, SD=24.14); t(18) = 2.65, p = 0.01. This might also be caused by the minimization of the displayed information, carried out by the system in the stress condition, similar to the case of the perceived SA*
- NASA-TLX frustration score, yielding no significant effect on perceived workload for stress; t(18) = 0.11, p = 0.9, in the two versions of the system studied

- UMUX-Lite results; yielding no significant effect on perceived UX for stress; t(18) = -0.2, p = 0.84, in the two versions of the system studied

In conclusion, the hypotheses *H4*, *H4b*, *H4c*, *H4d*, *H4f* and *H4e* were rejected, with no statistically significant differences between the stress and non-stress conditions. Only the hypotheses *H4a* were confirmed, highlighting that stress affects perceived SA, when using the system.

5.5.5 Impact of Professional Expertise

Aside from the impact of stress in our results, we also assessed the effect of professional expertise in the perceived and observed SA, the workload, and UX of the participants, both with and without using the system. In particular, our hypotheses were defined as follows:

H5. Participants' professional expertise constitutes a factor influencing all observed measures

H5a. Participants' professional expertise has an impact on perceived SA when using the system, in stress or no stress conditions

H5b. Participants' professional expertise has an impact on perceived SA when not using the system, in stress or no stress conditions

H5c. Participants' professional expertise has an impact on observed SA when using the system, in stress or no stress conditions

H5d. Participants' professional expertise has an impact on observed SA when not using the system, in stress or no stress conditions

H5e. Participants' professional expertise has an impact on perceived workload with the system, in stress or no stress conditions

H5f. Participants' professional expertise has an impact on perceived UX with the system, in stress or no stress conditions

The expertise score for each individual was calculated as follows. For each question on the background information questionnaire, regarding professional expertise in different domains (Questions 4, 5a, 5b, 5c, and 5d in Annex A), a score was assigned in the following manner:

- If no expertise at all: score 0
- If less than 5 years of expertise: score 1
- If 5 to 10 years of expertise: score 2
- If more than 10 years of expertise score 3

These scores were then summed for each individual, characterizing their expertise as *Low* if the total score was below 3, *Moderate* if it was between 3 and 7, and *High* if it was above 7.

This scoring approach led to 8 participants with *High* professional expertise 7 with *Moderate*, and 5 with *Low*. To test the effect of professional expertise on the measurements that were studied throughout the experiment, we carried out 2-factor ANOVA without replication, since the number of participants in each expertise category was unequal, acquiring the following results:

- For *perceived SA without using the system*, an effect size of 0.7128 was found, indicating that 71.28% of variance in the overall SART scores was explained by professional expertise ($F(17.17)=2.49$, $p=0.03$). Post hoc t-tests applying the Bonferroni correction yielded statistically significant differences in perceived SA between participants with high expertise ($M=15.43$, $SD= 5.69$) and participants with low professional expertise ($M=21.25$, $SD=2.76$)

- For *perceived SA when using the system*, no significant effect of professional expertise was found on the overall SART scores ($F(18.18)=0.5840$, $p=0.86$). This is an interesting finding compared to the effect of professional expertise on perceived SA *without* the system, suggesting that, when using the system, any differences on perceived SA are diminished.

- For *observed SA*, no significant effect of professional expertise was found on the overall SAGAT scores, when using the system ($F(19.19)=0.7225$, $p=0.76$), or without it ($F(19.19)=0.7468$, $p=0.73$).

- For *perceived workload*, an effect size of 0.9353 was found, indicating that 93.53% of variance in the NASA-TLX scores was explained by professional expertise ($F(18.18)=15.07$. $p<0.0001$). Post hoc t-tests applying the Bonferroni correction did not reveal any statistically significant differences between the three groups: high expertise vs moderate expertise, high expertise vs low expertise, moderate expertise vs low expertise.

- Regarding the *overall UX*, it yielded an effect size of 0.819 indicating that 81.9% of variance in the UMUX Lite scores was explained by professional expertise ($F(16.16)=4.60$. $p=0.002$). Post hoc t-tests applying the Bonferroni correction did not reveal any statistically significant differences between the three groups: high expertise vs moderate expertise, high expertise vs low expertise, moderate expertise vs low expertise.

In conclusion, the results of the analysis indicate that professional expertise can explain variance in perceived SA without the system, perceived workload, and perceived UX. Interestingly, professional expertise did not have any effect on observed SA (with or without the system) and perceived SA with the system. This led to the conclusion that the impact of the system on the LEAs' SA is not dependent on their field expertise, and as such it can provide the same benefits

for all users. This finding is even more important considering that professional expertise did have an effect on perceived workload and perceived UX, confirming that, despite any perceived issues with respect to workload or UX, the SA achieved with the system remains the same for all users, independently of their professional expertise.

5.5.6 Situational Awareness Levels

As already mentioned, the theoretical model of SA [45] involves 3 levels: perceiving critical factors in the scene (Level 1 SA), understanding their meaning (Level 2 SA), and predicting how they will evolve (Level 3 SA). In order to further examine differences across conditions for all SA levels, paired two-tailed t-tests were conducted. The results have as follows:

- For *Level 1 SA, in the stress condition*, a statistically significant difference was found between using the system (M=74.58, SD=40.6) and without the system (M=98.33, SD=7.45); t(19)=-2.80, p=0.01. This is an important finding of this study, highlighting that participants' observed SA was better in stress conditions without the system's UI in the case of level 1 SA. A possible conclusion would be that the system should avoid providing obvious or trivial information in highly stressful situations, unless it is accompanied by some additional information pertaining to higher SA levels. For instance, when an ally is detected, it is not necessary to highlight them if no additional information can be provided. Future studies will explore if this was an effect of the simulation or if it is also confirmed in in-situ studies. A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 1 SA in stress condition, without any observed effect (F(19, 19)=1.36, p=0.25).
- For *Level 1 SA, in the no stress condition*, a paired t-test was conducted to compare results with the system and without it, yielding no statistically significant difference (t(19)=0.37, p=0.71). A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 1 SA, in the no stress condition, without any observed effect (F(19, 19)=1.35, p=0.26).
- *For Level 2 SA, in the stress condition*, a paired t-test was conducted to compare results with the system and without it, yielding no statistically significant difference (t(19)=1.34, p=0.20). A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 2 SA, in the stress condition, without any observed effect (F(19, 19)=1.21, p=0.34).
- For *Level 2 SA, in the no stress condition*, a paired t-test was conducted to compare results with the system (M=68.49, SD=15.07) and without it

Table 5.12: SAGAT per level

	Mean	**Min**	**Max**	**Range**	**SD**	**95% CI**
Level 1						
System Stress	74.58	0	100	100	40.6	[55.58, 93.58]
No System Stress	98.33	66.67	100	33.33	7.45	[94.84, 100]
System No stress	87.92	0	100	100	24.25	[76.57, 99.27]
No System No Stress	85.42	33.33	33.33	66.67	21.61	[75.3, 95.53]
Level 2						
System Stress	66.03	44.44	100	55.56	15.07	[58.97, 73,08]
No System Stress	59.58	33.33	100	66.67	16.94	[51.66, 67.51]
System No stress	68.49	33.33	88.89	55.56	15.07	[61.44, 75.54]
No System No Stress	57.71	30	87.5	57.5	17.55	[49.49, 65.92]
Level 3						
System Stress	74.83	0	100	100	31.07	[60.29, 89.38]
No System Stress	68	0	100	100	29.98	[53.97, 82.03]
System No stress	57.33	0	100	100	36.64	[40.19, 74.48]
No System No Stress	54	0	100	100	39	[35.7, 72.3]

(M=57.71, SD=17.55), yielding statistically significant difference (t(19)=2.16, p=0.04). This finding sheds light to the particular conditions in which the system has more impact, leading to the conclusion that, when LEAs are not stressed, their observed level 2 SA is substantially increased. A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 2 SA, in the no stress condition, without any observed effect (F(19, 19)=1.16, p=0.38).

- For *Level 3 SA, in the stress condition*, a paired t-test was conducted to compare results with the system and without it, yielding no statistically significant difference (t(19)=0.55, p=0.59). A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 3 SA, in the stress condition, without any observed effect (F(19, 19)=0.19, p=0.1).

- For *Level 3 SA, in the no stress condition*, a paired t-test was conducted to compare results with the system and without it, yielding no statistically significant difference (t(19)=0.33, p=0.75). A 2-factor ANOVA was also carried out to assess the impact of professional expertise on observed level 3 SA, in the stress condition, without any observed effect (F(19, 19)=1.74, p=0.12).

Effect of stress Paired t-tests were carried out to also explore the effect of stress on observed SA for the three different SA Levels, resulting in the following:

- When using the system, no statistically significant differences were found between the two different stress conditions regarding observed SA at Level 1 (t(19)=-1.25, p=0.23).

- Without the system, a statistically significant difference was found between the stress (M=98.33, SD=7.45) and the no stress condition (M=85.42, SD=21.61) regarding observed SA at Level 1; t(19)=2.41, p=0.02. This is an interesting finding since participants' Level 1 SA in stress outperformed their Level 1 SA when in no stress, leading to the conclusion that increased stress led to increased Level 1 SA for LEAs.

- When using the system, no statistically significant differences were found between the two different stress conditions regarding observed SA at Level 2 (t(19)=-0.51, p=0.62).

- Without the system, no statistically significant differences were found between the two different stress conditions regarding observed SA at Level 2 (t(19)=0.31, p=0.76).

- When using the system, no statistically significant differences were found between the two different stress conditions regarding observed SA at Level 3 (t(19)=-1.94, p=0.06).

- Without the system, no statistically significant differences were found between the two different stress conditions regarding observed SA at Level 3 (t(19)=-1.04, p=0.31).

5.5.7 Correlations between measurements

A correlation analysis of all the measurements acquired in this study did not reveal any strong correlations between observed SA, perceived SA, perceived workload and perceived UX in either stress or non-stress conditions.

Table 5.13: Correlations between measurements for the Stress condition

	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
UMUX-Lite	1			
NASA-TLX	0.14	1		
SART	0.37	0.26	1	
SAGAT	-0.31	0.03	-0.1	1

Table 5.14: Correlations between measurements for the Non-Stress condition

	Attentional Demand	**Attentional Supply**	**Understanding**	**Situational Awareness**
UMUX-Lite	1			
NASA-TLX	-0.06	1		
SART	0.29	0.44	1	
SAGAT	-0.15	0.09	-0.07	1

5.6 Discussion

This user study aimed to assess the decision making system (DM) and the delivered GUI with regard to three key dimensions, namely Situational Awareness (SA), workload, and User Experience (UX). In addition, it explored if and how LEAs' stress and professional expertise have an impact on the aforementioned metrics.

With respect to Situational Awareness, the study examined perceived and observed SA with the aim to identify whether the system enhances LEAs' SA in stressful and non-stressful conditions. Overall, use of the system improved perceived and observed SA, by 25.63% and 9.25% respectively. In particular, in the

case of stressful conditions, perceived SA was improved by 30%, whereas observed SA by 3.95%. In non-stressful conditions, perceived SA was improved by 15.65%, while observed SA by 15%. Our results indicate that, when the system is used by LEAs who are not stressed, it becomes even more effective, maximizing its benefits. In any case, it is notable that, even under stress conditions, the system achieves to improve LEAs' SA.

Further statistical analysis of the results regarding perceived SA yielded useful findings. Participants perceived their SA to be improved when using the system, in comparison to viewing the videos without the system, an effect which was stronger under stress condition. Examining closer the reasons for the LEAs' higher perceived SA under stress conditions, it became apparent that in both conditions (stress and no stress), participants believed that their understanding with the system was enhanced. In fact, under stress condition, participants perceived their understanding to be substantially improved with the system. At the same time, in the no stress condition, when using the system, participants found that greater attentional demand was required (compared to the no system condition), yet additional attentional supply was provided. These findings were observed in the no stress condition as expected, since, in this case, the system provides additional textual information besides the icons. Although these observations were not statistically significant, the overall SART score - which is produced based on all the subscales - was lower in the no stress condition. At the same time, qualitative findings from the debriefing discussion confirmed that some participants perceived that sometimes too much information was displayed.

Analysis of the observed SA, however, resulted in the paradoxical finding that although participants perceived that they had to pay greater attention to the system in the no stress condition, their actual SA performance was substantially improved. Moreover, further analysis of the SAGAT responses per SA level, identified that this difference was statistically important for level 2 SA, which means that, in this case, the system assists them to achieve substantially better performance in comprehending the situation at hand. Since training for using the system was quite limited, as opposed to the actual training LEAs' would receive before using the system, participants' suggestions that information provided by the system is in some cases extraneous can be attributed to their limited prior exposure to the system. The importance of training was also stressed by participants during the debriefing session.

At the same time, in the no stress condition, although LEAs' observed SA was improved with the system, the difference was not statistically important. Further analysis of the SAGAT responses per SA level, revealed that for level 1 SA, participants performed better without the system. As a result, in stressful situations, the system should avoid providing obvious or trivial information (e.g. highlighting allies), unless it is accompanied by some additional data pertaining to higher SA levels (e.g. stress level of an ally), thus assisting agents to comprehend the situation or be able to make near future projections. This will be accommodated in future refinements of the DM.

With regard to perceived workload induced by the system, results highlighted compatibility with findings from studies in real policing tasks. As such, it can be concluded that the system does not induce workload to LEAs while using it. Additional studies will be conducted in the future, estimating workload with the system in field trials.

Concerning UX, findings were positive for all UX dimensions, namely overall UX, usefulness, and usability. Taking into account qualitative findings from the debriefing discussion, it turns out that the majority of participants (70%) stated that they would use the system in their everyday tasks. Participants identified that they liked the increased SA that the system offers and emphasized on particular information that they found useful, such as identification of foes and assessment of threats, information about weapons, details about the health status of injured civilians, and stress-related information about allies. Furthermore, it was highlighted that the icons used were clear and easy to understand. Dislikes of participants focused on the headset employed and the colour of identification rectangles. Although not directly relevant to the current aims and objectives of this evaluation, it is noted that the headset that will actually be employed in future field trials will be an AR headset, thus this concern will be alleviated. As far as the identification rectangles are concerned, the final system will identify individuals by a highlighter contouring the person's body, thus occupying less screen real estate.

The limitations of the current user study are threefold. One potential limitation is related to the nature of the study being carried out as a simulation through videos - and not in-situ. However, this approach was selected, because it ensured a more controlled study environment, allowing us to replicate the exact same scenarios for all the participants. An in situ evaluation in the context of policing tasks would introduce uncertainty, unpredictability and high variability, making any generalizations and high level conclusions invalid. Nevertheless, future evaluations of the system will be carried out in-situ, for the invaluable qualitative findings that may occur. A second limitation pertains to the number of participants, which should be larger in order to provide greater confidence in the findings. Taking into account the structured approach that was followed and that the study can be easily repeated, future work will focus on replicating the study with additional end users. Finally, a third limitation of this study is that participants did not receive proper training on the system, thus sometimes they felt uncertain about the meaning of specific icons. Nevertheless, it is very encouraging that even without training the system was proved to be effective; it is expected that in real world usage, when agents receive adequate training prior to using it, its benefits can be maximized.

Chapter 6

Conclusions & Future Work

In this work, we introduced a novel computational methodology, which aims at enhancing the SA of users, through a real-time, dynamic adaptation of UIs, while taking into consideration the current context. Our approach combines Combinatorial Optimization with Ontology modeling and reasoning in order to graphically provide suitable information at run-time, through deciding *what* information to present, *when* to present it, *where* to visualize it in the display, and *how*. This is performed while considering placement constraints of GUI elements, as well as avoiding prominent "SA daemons", such as information overload and induced stress. In this respect, a Convolutional Neural Network (CNN) was also developed for the stress detection and 3-level stress classification tasks, achieving state-of-the-art results. In particular the obtained accuracy for the stress detection task is 98.6%, while for the 3-level stress classification task is 85.3 %.

In the context of this thesis, we deployed the proposed, general-purpose methodology to the application domain of the *DARLENE* project, whose main objective is to improve the SA of Law Enforcement Agents (LEAs), when responding to criminal and terrorist activities, through Augmented Reality and Machine Learning technologies. The proposed computational approach aims to aid LEAs in making more informed and rapid decisions, through in-situ dynamic adaptivity of the visual elements that are presented on their AR headsets, taking into account the variety of user characteristics, environmental and system factors, as well as the current task. For the purpose of identifying these factors that affect LEA's SA, as well as GUI elements that would increase their SA during policing, co-creation workshops were conducted with end-users. The requirements that resulted from these workshops enabled us to model knowledge from this application domain into an Ontology and formulate an optimization problem for the adaptation of the LEA's AR UI.

To assess our methodology, two evaluations were conducted, proving us with invaluable insight, with respect to the benefits and limitations of our approach. The first one was an expert-based evaluation with 10 LEAs and User Experience (UX) experts, assessing the appropriateness of the system's decisions, regarding what information was displayed, how detailed it was and where it was positioned.

The results led to improvements in both the positioning and presentation of the GUI elements, which were employed in the subsequent evaluation with end-users.

In that second, user-based evaluation, 20 LEAs from different agencies were involved. Its aim was to assess our approach and its adaptive capabilities with regard to three key dimensions, namely SA, workload, and User Experience (UX). Acknowledging the influence of stress in SA, these metrics were evaluated both at normal stress states and under experimentally induced stress. In addition, it was explored if and how LEAs' stress and professional expertise have an impact on the aforementioned metrics. With respect to Situational Awareness, the study examined perceived and observed SA with the aim to identify whether the system enhances LEAs' SA in stressful and non-stressful conditions. Overall, using the system improved perceived and observed SA, by 25.63% and 9.25% respectively. In particular, in the case of stressful conditions, perceived SA was improved by 30%, whereas observed SA by 3.95%. In non-stressful conditions, perceived SA was improved by 15.65%, while observed SA by 15%. Furthermore, the results indicate that the system does not induce perceived workload, in both conditions, when compared with findings from studies in real policing tasks, and that it is both useful and usable, providing an overall positive UX. Lastly, when using the system, professional expertise did not have an effect on observed or perceived SA, indicating that our system can benefit anyone, while stress negatively influenced perceived effort, in comparison to the no stress condition, but influenced positively perceived SA. Although the finding that in the no stress condition participants exhibited lower perceived SA may seem unconventional, it is noted that perceived SA is calculated by taking into account the attentional demand required, which was deemed as higher in the no stress condition, since the GUI included more information. It is noteworthy that in this case, observed SA was substantially improved, despite participants' scoring lower in perceived SA.

Regarding the claimed contribution and comparative strengths of the proposed methodology, we introduced a novel approach for the generation and adaptation of GUIs, dynamically at run-time, which combines Ontology modeling and reasoning with Combinatorial Optimization. The parameters of the optimization problem are not static or manually defined by content creators, but dynamically inferred, based on the user's stress and other factors of the current situation. Furthermore, beyond the existing literature, all dimensions of the visualization problem (what, when, how, where) are simultaneously formulated in the optimization, and jointly solved, resulting in an improved, holistic handling. At the same time, the layout of the UI is dynamically defined, with the positions of the graphical elements being defined at run-time, based on the current scene. We also developed a stress classification model with an accuracy of 85.3%, which surpasses the state-of-the-art for this dataset, at 83%. Moreover, we presented a methodology that can improve the Situational Awareness of Law Enforcement Agents, while not inducing workload and providing a positive user experience. To the best of our knowledge, this is a first attempt in this area. Finally, we introduced a systematic methodology for evaluating systems that aim at enhancing the SA, which includes experts and a

user-based study.

With regard to future directions, a first step is to address the issues discovered during the expert and user-based evaluations. One of the aspects we seek to improve in future versions is the better accommodation of the LEA's stress. Concretely, although the system enhances observed SA in all states, even under stressful conditions, its benefit is not as emphatic as in normal stress conditions, as indicated by our analysis. Moreover, with regard to future experiments, we plan to evaluate our system in in-situ simulations, with a larger pool of participants, consolidating our results and acquiring new findings that did not arise in our video-simulation approach.

Furthermore, the evaluations noted certain inconsistencies in preferences across participants, which need to be explored in more detail. To this end, we plan to accommodate further customization and personalization in our approach, through enhanced user-modeling and incorporation of advanced content-recommendation techniques.

Moreover, a limitation in our modeling is that it takes into account only the type of information and not its content. Consequently, as an example, both a knife and an explosive will be considered equally important for enhancing the user's SA by our optimization algorithm, since they are both "Carried Weapons". We plan to address this by modeling, and considering in the optimization formulation, a "criticality" attribute that would aid in differentiating between different levels of priority, for the same kind of information.

In addition, as we have already established, given the complexity of our optimization problem and the real-time requirements for its deployment, achieving fine-grained pixel placement of GUI elements without "down-sampling" of the display, and, at the same time, having fine-grained collision detection, is computationally intractable, using off-the-shelf optimizers. In this respect, we have already started investigating the adoption of Machine Learning methods for Combinatorial Optimization, a new prominent area of research in recent years. Considering that our visualization problem is an instance of the 0-1 Knapsack problem, which is NP-complete, such a direction could substantially enhance the scalability of our approach, allowing us at the same time to incorporate more complex constraints and improve placement, without resorting to coarse collision detection.

Bibliography

[1] Chapter Three - RDF and the Semantic Web Stack. In Olivier Curé and Guillaume Blin, editors, *RDF Database Systems*, pages 41–80. Morgan Kaufmann, Boston, January 2015.

[2] A. F. Norcio and J. Stanley. Adaptive human-computer interfaces: A literature survey and perspective. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(2):399–408, March 1989.

[3] Sunitha Abburu and Suresh Babu Golla. Ontology Storage Models and Tools: An Authentic Survey. *Journal of Intelligent Systems*, 25(4):539–553, October 2016.

[4] Sajjad Abdoli, Patrick Cardinal, and Alessandro Lameiras Koerich. End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network. *arXiv:1904.08990 [cs, stat]*, April 2019.

[5] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In Hans-W. Gellersen, editor, *Handheld and Ubiquitous Computing*, pages 304–307, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[6] Gregory D Abowd, Anind K Dey, Robert Orr, and Jason Brotherton. Context-Awareness in Wearable and Ubiquitous Computing. page 3.

[7] Jose Aguilar, Marxjhony Jerez, and Taniana Rodríguez. CAMeOnto: Context awareness meta ontology modeling. *Applied Computing and Informatics*, 14(2):202–213, July 2018.

[8] Ahmet Akbas. Evaluation of the physiological data indicating the dynamic stress level of drivers. *Scientific Research and Essays*, 6(2):430–439, January 2011.

[9] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. Adaptive Model-Driven User Interface Development Systems. *ACM Computing Surveys*, 47(1):9:1–9:33, May 2014.

[10] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. Engineering Adaptive Model-Driven User Interfaces. *IEEE Transactions on Software Engineering*, 42(12):1118–1147, December 2016.

[11] Fadi Al Machot, Heinrich C. Mayr, and Suneth Ranasinghe. A Hybrid Reasoning Approach for Activity Recognition Based on Answer Set Programming and Dempster–Shafer Theory. In Kyandoghere Kyamakya, Wolfgang Mathis, Ruedi Stoop, Jean Chamberlain Chedjou, and Zhong Li, editors, *Recent Advances in Nonlinear Dynamics and Synchronization: With Selected Applications in Electrical Engineering, Neurocomputing, and Transportation*, Studies in Systems, Decision and Control, pages 303–318. Springer International Publishing, Cham, 2018.

[12] Unai Alegre, Juan Carlos Augusto, and Tony Clark. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, 117:55–83, July 2016.

[13] Konstantinos C. Apostolakis, Nikolaos Dimitriou, George Margetis, Stavroula Ntoa, Dimitrios Tzovaras, and Constantine Stephanidis. DARLENE–Improving situational awareness of European law enforcement agents through a combination of augmented reality and artificial intelligence solutions. *Open Research Europe*, 1(87):87, 2021.

[14] Mark Assad, David J. Carmichael, Judy Kay, and Bob Kummerfeld. PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive Computing*, pages 55–72, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[15] Ehsan Azimi, Camilo Molina, Alexander Chang, Judy Huang, Chien-Ming Huang, and Peter Kazanzides. Interactive Training and Operation Ecosystem for Surgical Tasks in Mixed Reality. In Danail Stoyanov, Zeike Taylor, Duygu Sarikaya, Jonathan McLeod, Miguel Angel González Ballester, Noel C.F. Codella, Anne Martel, Lena Maier-Hein, Anand Malpani, Marco A. Zenati, Sandrine De Ribaupierre, Luo Xiongbiao, Toby Collins, Tobias Reichl, Klaus Drechsler, Marius Erdt, Marius George Linguraru, Cristina Oyarzun Laura, Raj Shekhar, Stefan Wesarg, M. Emre Celebi, Kristin Dana, and Allan Halpern, editors, *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Lecture Notes in Computer Science, pages 20–29, Cham, 2018. Springer International Publishing.

[16] B. Kurz, I. Popescu, and S. Gallacher. FACADE - a framework for context-aware content adaptation and delivery. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, pages 46–55, May 2004.

[17] Zahra Bahramian, Rahim Ali Abbaspour, and Christophe Claramunt. A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning. *Mobile Information Systems*, 2017:e9364903, September 2017.

[18] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. MenuOptimizer: Interactive optimization of menu systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 331–342, St. Andrews Scotland, United Kingdom, October 2013. ACM.

[19] Anjana Bali and Amteshwar Singh Jaggi. Clinical experimental stress studies: Methods and assessment. *Reviews in the Neurosciences*, 26(5):555–579, October 2015.

[20] Lionel Balme, Alexandre Demeure, Nicolas Barralon, Joëlle Coutaz, and Gaëlle Calvary. CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In Panos Markopoulos, Berry Eggen, Emile Aarts, and James L. Crowley, editors, *Ambient Intelligence*, Lecture Notes in Computer Science, pages 291–302, Berlin, Heidelberg, 2004. Springer.

[21] Joseph Bauer, Ralf-Detlef Kutsche, and Rudiger Ehrmanntraut. Identification and modeling of contexts for different information scenarios in air traffic. *Technische Universität Berlin, Diplomarbeit*, 62, 2003.

[22] Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 101–110, New York, NY, USA, November 2001. Association for Computing Machinery.

[23] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, April 2010. support for reasoning for every model.

[24] Antonis Bikakis, Theodore Patkos, Grigoris Antoniou, and Dimitris Plexousakis. A Survey of Semantics-Based Approaches for Context Reasoning in Ambient Intelligence. In Max Mühlhäuser, Alois Ferscha, and Erwin Aitenbichler, editors, *Constructing Ambient Intelligence*, volume 11, pages 14–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[25] Willem Nico Borst and W. N. Borst. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. September 1997.

[26] Oliver Brdiczka, James L. Crowley, and Patrick Reignier. Learning Situation Models in a Smart Home. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):56–63, February 2009.

[27] Joseph L. Brooks. Counterbalancing for serial order carryover effects in experimental condition orders. *Psychological Methods*, 17(4):600–614, December 2012.

[28] P.J. Brown, J.D. Bovey, and Xian Chen. Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58–64, October 1997.

[29] Thomas Butter, Markus Aleksy, Philipp Bostan, and Martin Schader. Context-aware User Interface Framework for Mobile Applications. In *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, pages 39–39, Toronto, ON, Canada, 2007. IEEE.

[30] Tomas Cerny, Karel Cemus, Michael J. Donahoo, and Eunjee Song. Aspect-driven, data-reflective and context-aware user interfaces design. *ACM SIGAPP Applied Computing Review*, 13(4):53–66, December 2013.

[31] Harry Chen. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. December 2004.

[32] Harry Chen, Tim Finin, and Anupam Joshi. Using OWL in a Pervasive Computing Broker. *Workshop on Ontologies in Agent Systems, AAMAS-2003*, July 2003.

[33] Harry Chen, Tim Finin, and Anupam Joshi. Semantic Web in a Pervasive Context-Aware Architecture. Technical report, MARYLAND UNIV BALTIMORE, January 2005.

[34] Peter Pin-Shan Chen. The entity-relationship model-toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976.

[35] Keith Cheverst, Keith Mitchell, and Nigel Davies. Design of an object model for a context sensitive tourist GUIDE. *Computers & Graphics*, 23(6):883–891, December 1999.

[36] Hyun-Myung Cho, Heesu Park, Suh-Yeon Dong, and Inchan Youn. Ambulatory and laboratory stress detection based on raw electrocardiogram signals using a convolutional neural network. *Sensors*, 19(20):4408, January 2019.

[37] Hao-hua Chu, Henry Song, Candy Wong, Shoji Kurakake, and Masaji Katagiri. Roam, a seamless application framework. *Journal of Systems and Software*, 69(3):209–226, January 2004.

[38] Klen Čopič Pucihar and Matjaž Kljun. ART for Art: Augmented Reality Taxonomy for Art and Cultural Heritage. In Vladimir Geroimenko, editor, *Augmented Reality Art: From an Emerging Technology to a Novel Creative Medium*, Springer Series on Cultural Computing, pages 73–94. Springer International Publishing, Cham, 2018.

[39] Joost de Winter, P.M. Leeuwen, and Riender Happee. *Advantages and Disadvantages of Driving Simulators: A Discussion*. January 2012.

[40] Mahsa Dehghani Soufi, Taha Samad-Soltani, Samad Shams Vahdati, and Peyman Rezaei-Hachesu. Decision support system for triage management: A hybrid approach using rule-based reasoning and fuzzy logic. *International Journal of Medical Informatics*, 114:35–44, June 2018.

[41] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human–Computer Interaction*, 16(2-4):97–166, December 2001.

[42] Anind K Dey, Gregory D Abowd, and Andrew Wood. CyberDesk: A framework for providing self-integrating context-aware services. page 11, 1998.

[43] S. DiVerdi, T. Hollerer, and R. Schreyer. Level of Detail Interfaces. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 300–301, Arlington, VA, USA, 2004. IEEE.

[44] Mitchell Doughty, Karan Singh, and Nilesh R. Ghugre. SurgeonAssist-Net: Towards Context-Aware Head-Mounted Display-Based Augmented Reality for Surgical Guidance. *arXiv:2107.06397 [cs]*, July 2021. Comment: Accepted at MICCAI 2021; 11 pages, 3 figures.

[45] Mica Endsley. Endsley, M.R.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors Journal 37(1), 32-64. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37:32–64, March 1995.

[46] Mica R. Endsley. Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors Society Annual Meeting*, 32(2):97–101, October 1988.

[47] Mica R. Endsley. SA demons: The enemies of situation awareness. In *Designing for Situation Awareness*, pages 50–61. CRC Press, 2016.

[48] Mica R Endsley. *Direct Measurement of Situation Awareness: Validity and Use of SAGAT*, pages 129–156. Routledge, first edition, July 2017. - Therefore a random selection from a constant set of queries is recommended at each stop.

- static knowledge, such as procedures or rules for performing tasks, is outside the bounds of an SA requirements analysis. The analysis focuses only on the dynamic situational information that affects what the operators do.

[49] Mica R. Endsley, Betty Bolte, and Debra G. Jones. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, London, July 2003.

[50] M.R. Endsley. Situation awareness global assessment technique (SAGAT). In *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, pages 789–795, Dayton, OH, USA, 1988. IEEE.

[51] World Leaders in Research-Based User Experience. Success Rate: The Simplest Usability Metric. https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/.

[52] Andreas Fender, Philipp Herholz, Marc Alexa, and Jörg Müller. OptiSpace: Automated Placement of Interactive 3D Projection Mapping Content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, Montreal QC Canada, April 2018. ACM.

[53] Gerhard Fischer. Context-aware systems: The 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*, page 287, Capri Island, Italy, 2012. ACM Press.

[54] Viking Forsman. *MEASURING SITUATION AWARENESS IN MIXED REALITY SIMULATIONS*. 2019.

[55] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, 2004.

[56] David Franklin and Joshua Flaschsbart. All Gadget and No Representation Makes Jack a Dull Environment. page 6.

[57] G. Kortuem, Z. Segall, and M. Bauer. Context-aware, adaptive wearable computers as remote interfaces to 'intelligent' environments. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, pages 58–65, 19.

[58] Krzysztof Gajos and Daniel S Weld. SUPPLE: Automatically Generating User Interfaces. page 8.

[59] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '06*, page 201, Venezia, Italy, 2006. ACM Press.

[60] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 231–240, New York, NY, USA, October 2007. Association for Computing Machinery.

[61] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. FLARE: Fast layout for augmented reality applications. In *2014 IEEE International Symposium*

on Mixed and Augmented Reality (ISMAR), pages 207–212, Munich, Germany, September 2014. IEEE.

[62] Damianos Gavalas, Stella Sylaiou, Vlasios Kasapakis, and Elena Dzardanova. Special issue on virtual and mixed reality in culture and heritage. *Personal and Ubiquitous Computing*, 24(6):813–814, December 2020.

[63] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Transactions on Computer-Human Interaction*, 24(2):14:1–14:33, April 2017.

[64] Nehla Ghouaiel, Jean-Marc Cieutat, and Jean-Pierre Jessel. Adaptive augmented reality: Plasticity of augmentations. In *Proceedings of the 2014 Virtual Reality International Conference*, VRIC '14, pages 1–4, New York, NY, USA, April 2014. Association for Computing Machinery.

[65] Giorgos Giannakakis, Dimitris Grigoriadis, Katerina Giannakaki, Olympia Simantiraki, Alexandros Roniotis, and Manolis Tsiknakis. Review on psychological stress detection using biosignals. *IEEE Transactions on Affective Computing*, pages 1–1, 2019. Normally, HRV parameters need at least 5 minutes in order to exist enough samples [215], however, there are some studies claiming that even with a shorter time period, can provide reliable estimators of mental stress [210, 216].

The R-peak is the most prominent and most of the anal-yses exploit the distribution this peak through successive R peaks intervals, or RR intervals (RRI).

This measure [HR] is the most straightforward and the most used with the vast majority being a reliable measure for the arousal part of stress.

[66] Giorgos Giannakakis, Eleftherios Trivizakis, Manolis Tsiknakis, and Kostas Marias. A novel multi-kernel 1D convolutional neural network for stress recognition from ECG. In *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 1–4, September 2019.

[67] Jorge Marx Gómez and Thanh Tran. A Survey on Approaches to Adaptation on the Web. 2009.

[68] Philip Gray and Daniel Salber. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In Murray Reed Little and Laurence Nigay, editors, *Engineering for Human-Computer Interaction*, Lecture Notes in Computer Science, pages 317–335, Berlin, Heidelberg, 2001. Springer.

[69] T. Gruber. A translation approach to portable ontology specifications. *undefined*, 1993.

[70] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human-Computer Studies*, pages 907–928. Kluwer Academic Publishers, 1993.

[71] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1706–1724, June 2017.

[72] Tao Gu, Xiao Wang, and Daqing Zhang. An Ontology-based Context Model in Intelligent Environments. *CNDS 2004*, January 2004.

[73] Naser Hakimi, Ata Jodeiri, Mahya Mirbagheri, and S. Kamaledin Setarehdan. Proposing a convolutional neural network for stress assessment by means of derived heart rate from functional near infrared spectroscopy. *Computers in Biology and Medicine*, 121:103810, June 2020.

[74] T. A. Halpin. *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design.* Morgan Kaufmann Series in Data Management Systems. Morgan Kaufman Publishers, San Francisco, 2001. Industrial foreword – Academic foreword – Preface – 1 Introduction – 2 Information levels and frameworks – 3 Conceptual modeling: first steps – 4 Uniqueness constraints – 5 Mandatory roles – 6 Value, set comparison and subtype constraints – 7 Other constraints and final checks – 8 Entity relationship modeling – 9 Data modeling in UML – 10 Relational mapping – 11 Relational languages – 12 Schema transformations – 13 Other modeling aspects and trends – Answers – Bibliography – ORM glossary – ER glossary – UML glossary – Index.

[75] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Peter A. Hancock and Najmedin Meshkati, editors, *Advances in Psychology*, volume 52 of *Human Mental Workload*, pages 139–183. North-Holland, January 1988.

[76] Jiayuan He, Ke Li, Xiaoli Liao, Ping Zhang, and Ning Jiang. Real-time detection of acute cognitive stress using a convolutional neural network from electrocardiographic signal. *IEEE Access*, 7:42710–42717, 2019.

[77] J.A. Healey and R.W. Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, June 2005.

[78] Jennifer Anne Healey. *Wearable and Automotive Systems for Affect Recognition from Physiology.* Thesis, Massachusetts Institute of Technology, 2000.

[79] K. Henricksen and J. Indulska. Modelling and using imperfect context information. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, pages 33–37, March 2004.

[80] Karen Henricksen. A framework for context-aware pervasive computing applications. 2003.

[81] Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, February 2006.

[82] Karen Henricksen, Jadwiga Indulska, and Ted McFadden. Modelling Context Information with ORM. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, Lecture Notes in Computer Science, pages 626–635, Berlin, Heidelberg, 2005. Springer.

[83] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling Context Information in Pervasive Computing Systems. In Friedemann Mattern and Mahmoud Naghshineh, editors, *Pervasive Computing*, Lecture Notes in Computer Science, pages 167–180, Berlin, Heidelberg, 2002. Springer.

[84] Tasha Hollingsed and David G. Novick. Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th Annual ACM International Conference on Design of Communication*, SIGDOC '07, pages 249–255, New York, NY, USA, October 2007. Association for Computing Machinery.

[85] M. Robiul Hoque, M Humayun Kabir, Kshav Thappa, and Sung-Hyun Yang. Ontology-based Context Modeling to Facilitate Reasoning in a Context-Aware System: A Case Study for the Smart Home. *International Journal of Smart Home*, 9:151–156, September 2015.

[86] Matthew Horridge. A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.3. page 109.

[87] Jing Huang, Xiong Luo, and Xiaoyan Peng. A novel classification method for a driver's cognitive stress level by transferring interbeat intervals of the ECG signal to pictures. *Sensors*, 20(5):1340, January 2020.

[88] Shu-Hsien Huang, Ting-Ting Wu, Hui-Chun Chu, and Gwo-Jen Hwang. A Decision Tree Approach to Conducting Dynamic Assessment in a Context-Aware Ubiquitous Learning Environment. In *Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (Wmute 2008)*, pages 89–94, March 2008.

[89] Jamil Hussain, Anees Ul Hassan, Hafiz Syed Muhammad Bilal, Rahman Ali, Muhammad Afzal, Shujaat Hussain, Jaehun Bang, Oresti Banos, and Sungyoung Lee. Model-based adaptive user interface based on context and user experience evaluation. *Journal on Multimodal User Interfaces*, 12(1):1–16, March 2018.

[90] Bosun Hwang, Jiwoo You, Thomas Vaessen, Inez Myin-Germeys, Cheolsoo Park, and Byoung-Tak Zhang. Deep ECGNet: An optimal deep learning framework for monitoring mental stress using ultra short-term ECG signals. *Telemedicine and e-Health*, 24(10):753–772, October 2018.

[91] Jadwiga Indulska, Ricky Robinson, Andry Rakotonirainy, and Karen Henricksen. Experiences in Using CC/PP in Context-Aware Systems. In Ming-Syan Chen, Panos K. Chrysanthis, Morris Sloman, and Arkady Zaslavsky, editors, *Mobile Data Management*, Lecture Notes in Computer Science, pages 247–261, Berlin, Heidelberg, 2003. Springer.

[92] Vivien Johnston, Michaela Black, Jonathan Wallace, Maurice Mulvenna, and Raymond Bond. A Framework for the Development of a Dynamic Adaptive Intelligent User Interface to Enhance the User Experience. In *Proceedings of the 31st European Conference on Cognitive Ergonomics*, pages 32–35, BELFAST United Kingdom, September 2019. ACM.

[93] Mingu Kang, Siho Shin, Jaehyo Jung, and Youn Tae Kim. Classification of mental stress using CNN-LSTM algorithms with electrocardiogram signals. *Journal of Healthcare Engineering*, 2021:e9951905, June 2021.

[94] Darko Katić, Patrick Spengler, Sebastian Bodenstedt, Gregor Castrillon-Oberndorfer, Robin Seeberger, Juergen Hoffmann, Ruediger Dillmann, and Stefanie Speidel. A system for context-aware intraoperative augmented reality in dental implant surgery. *International Journal of Computer Assisted Radiology and Surgery*, 10(1):101–108, January 2015.

[95] Hannes Kaufmann and Dieter Schmalstieg. Mathematics and geometry education with collaborative augmented reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, pages 37–41, New York, NY, USA, July 2002. Association for Computing Machinery.

[96] Mehmet Kesim and Yasin Ozarslan. Augmented Reality in Education: Current Technologies and the Potential for Education. *Procedia - Social and Behavioral Sciences*, 47:297–302, January 2012.

[97] Eunseok Kim, Jungwha Kim, and Woontack Woo. Metadata schema for context-aware augmented reality applications in cultural heritage domain. In *2015 Digital Heritage*, volume 2, pages 283–290, September 2015.

[98] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[99] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, April 2021.

[100] Sarah Krings, Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer, and Gregor Engels. Development framework for context-aware augmented reality applications. In *Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 1–6, Sophia Antipolis France, June 2020. ACM.

[101] Jaewon Lee, Hyeonjeong Lee, and Miyoung Shin. Driving stress detection using multimodal convolutional neural networks with nonlinear representation of short-term physiological signals. *Sensors*, 21(7):2381, January 2021.

[102] James R. Lewis, Brian S. Utesch, and Deborah E. Maher. UMUX-LITE: When there's no time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2099–2102, New York, NY, USA, April 2013. Association for Computing Machinery.

[103] Xin Li, José-Fernán Martínez, and Gregorio Rubio. Towards a Hybrid Approach to Context Reasoning for Underwater Robots. *Applied Sciences*, 7(2):183, February 2017.

[104] H. Lieberman and T. Selker. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3.4):617–632, 2000.

[105] Brian Y. Lim and Anind K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pages 13–22, 2010.

[106] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 147–160, New Orleans LA USA, October 2019. ACM.

[107] Víctor López-Jaquero, Jean Vanderdonckt, Francisco Montero, and Pascual González. Towards an Extended Model of User Interface Adaptation: The Isatine Framework. In Jan Gulliksen, Morton Borup Harning, Philippe Palanque, Gerrit C. van der Veer, and Janet Wesson, editors, *Engineering Interactive Systems*, pages 374–392, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[108] Eduardo Lupiani, Jose M. Juarez, Jose Palma, and Roque Marin. Monitoring elderly people at home with temporal Case-Based Reasoning. *Knowledge-Based Systems*, 134:116–134, October 2017.

[109] Delfina Malandrino, Francesca Mazzoni, Daniele Riboni, Claudio Bettini, Michele Colajanni, and Vittorio Scarano. MIMOSA: Context-aware adaptation for ubiquitous web access. *Personal and Ubiquitous Computing*, 14(4):301–320, May 2010.

[110] Silvano Martello. Knapsack problems: Algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimiza tion*, 1990.

[111] John McCarthy. Notes on formalizing context. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence - Volume 1*, IJCAI'93, pages 555–560, San Francisco, CA, USA, August 1993. Morgan Kaufmann Publishers Inc.

[112] John McCarthy and Sasa Buvac. Formalizing Context (Expanded Notes). *CSLI Lecture Notes*, 81:13–50, 1998.

[113] George A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63(2):81–97, 1956.

[114] Moody, George, Pollard, Tom, and Moody, Benjamin. WFDB software package. PhysioNet.

[115] Vivian Genaro Motti and Jean Vanderdonckt. A computational framework for context-aware adaptation of user interfaces. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12, Paris, France, May 2013. IEEE.

[116] Shane T. Mueller and Brian J. Piper. The Psychology Experiment Building Language (PEBL) and PEBL Test Battery. *Journal of neuroscience methods*, 222:250–259, January 2014.

[117] Jeffrey Nichols, Duen Horng Chau, and Brad A. Myers. Demonstrating the viability of automatically generated user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292, 2007.

[118] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. SnapToReality: Aligning Augmented Reality to the Real World. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1233–1244, San Jose California USA, May 2016. ACM.

[119] Petteri Nurmi and Patrik Floréen. Reasoning in context-aware systems. *Helsinki Institute for Information Technology, Position paper*, 2004.

[120] Sejin Oh, Woontack Woo, Gist U.-vr Lab, and S. Korea. CAMAR: Context-aware Mobile Augmented Reality in Smart Space.

[121] Tal Oron-Gilad, James Szalma, Shawn Stafford, and Peter Hancock. The Workload and Performance Relationship in the Real World: A Study of Police Officers in a Field Shooting Exercise. *International journal of occupational safety and ergonomics : JOSE*, 14:119–31, February 2008.

[122] Antti Oulasvirta. User Interface Design with Combinatorial Optimization. *Computer*, 50(1):40–47, January 2017.

[123] Antti Oulasvirta, Niraj Ramesh Dayama, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. Combinatorial Optimization of Graphical User Interface Designs. *Proceedings of the IEEE*, 108(3):434–464, March 2020.

[124] J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, pages 92–99, Pittsburgh, PA, USA, 1998. IEEE Comput. Soc.

[125] Mike Perkowitz and Oren Etzioni. Towards adaptive Web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1):245–275, April 2000.

[126] Mikko Perttunen, Jukka Riekki, and Ora Lassila. Context Representation and Reasoning in Pervasive Computing: A Review. *International Journal of Multimedia and Ubiquitous Engineering*, 4(4):28, 2009. on rule-based, logic programming, ontology-based

(description logic), and case-based reasoning.

[127] Preeja Pradeep and Shivsubramani Krishnamoorthy. The MOM of context-aware systems: A survey. *Computer Communications*, 137:44–69, March 2019.

[128] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. Towards an Extensible Context Ontology for Ambient Intelligence. In Panos Markopoulos, Berry Eggen, Emile Aarts, and James L. Crowley, editors, *Ambient Intelligence*, Lecture Notes in Computer Science, pages 148–159, Berlin, Heidelberg, 2004. Springer.

[129] Thanyalak Rattanasawad, Kanda Runapongsa Saikaew, Marut Buranarach, and Thepchai Supnithi. A review and comparison of rule languages and rule-based inference engines for the Semantic Web. In *2013 International Computer Science and Engineering Conference (ICSEC)*, pages 1–6, September 2013.

[130] Bryan R. Rill and Matti M. Hämäläinen. *The Art of Co-Creation: A Guidebook for Practitioners*. Springer, August 2018.

[131] Dairazalia Sanchez, Monica Tentori, and Jesus Favela. Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments. In *Eighth Mexican International Conference on Current Trends in Computer Science (ENC 2007)*, pages 33–40, September 2007.

[132] Sayan Sarcar, Jussi P.P. Jokinen, Antti Oulasvirta, Zhenxin Wang, Chaklam Silpasuwanchai, and Xiangshi Ren. Ability-Based Optimization of Touchscreen Interactions. *IEEE Pervasive Computing*, 17(1):15–26, January 2018.

[133] Bill N Schilit, Norman Adams, and Roy Want. Context-Aware Computing Applications. page 7, 1994.

[134] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32, September 1994.

[135] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced Interaction in Context. In Hans-W. Gellersen, editor, *Handheld and Ubiquitous Computing*, Lecture Notes in Computer Science, pages 89–101, Berlin, Heidelberg, 1999. Springer.

[136] Stephen J. Selcon and R. M. Taylor. Evaluation of the Situational Awareness Rating Technique(SART) as a tool for aircrew systems design. *AGARD, Situational Awareness in Aerospace Operations 8 p(SEE N 90-28972 23-53)*, 1990.

[137] Wonju Seo, Namho Kim, Sehyeon Kim, Chanhee Lee, and Sung-Min Park. Deep ECG-Respiration network (DeepER net) for recognizing mental stress. *Sensors*, 19(13):3021, January 2019.

[138] Nandita Sharma and Tom Gedeon. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer Methods and Programs in Biomedicine*, 108(3):1287–1301, December 2012.

[139] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556 [cs]*, April 2015.

[140] Nikolaos Simou and Stefanos D. Kollias. FiRE : A fuzzy reasoning engine for imprecise knowledge. 2007.

[141] Nikos Simou, Thanos Athanasiadis, Vassilis Tzouvaras, and Stefanos Kollias. Multimedia Reasoning with f-SHIN. In *Second International Workshop on Semantic Media Adaptation and Personalization (SMAP 2007)*, pages 44–49, December 2007.

[142] Makram Soui, Soumaya Diab, Ali Ouni, Aroua Essayeh, and Mourad Abed. An Ontology-Based Approach for User Interface Adaptation. volume 512, pages 199–215. January 2017.

[143] Giorgos Stoilos, Giorgos Stamou, Vassilis Tzouvaras, Jeff Pan, and Ian Horrocks. *Fuzzy OWL: Uncertainty and the Semantic Web.* January 2005.

[144] Thomas Strang and Claudia Linnhoff-Popien. A Context Modeling Survey. page 8. Only models.

[145] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1):161–197, March 1998.

[146] Robert Stutely. The Standard Generalized Markup Language. In Rae A. Earnshaw, editor, *Workstations and Publication Systems*, pages 176–189. Springer, New York, NY, 1987.

[147] Tom N. Tombaugh. A comprehensive review of the paced auditory serial addition test (PASAT). *Archives of clinical neuropsychology*, 21(1):53–76, 2006.

[148] Konstantinos Tzevelekakis, Zinovia Stefanidi, and George Margetis. Real-time stress level feedback from raw ECG signals for personalised, context-aware applications using lightweight convolutional neural network architectures. *Manuscript submitted for publication.*, September 2021.

[149] W. A. G. UAProf. Wireless Application Group–User Agent Profile Specification. In *WAP Forum, November*, volume 10, 1999.

[150] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(2):93–136, June 1996.

[151] Kristof Van Laerhoven. Combining the Self-Organizing Map and K-Means Clustering for On-Line Classification of Sensor Data. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, Lecture Notes in Computer Science, pages 464–469, Berlin, Heidelberg, 2001. Springer.

[152] Xiao Hang Wang, Tao Gu, Da Qing Zhang, and Hung Keng Pung. Ontology Based Context Modeling and Reasoning using OWL. In *In: IEEE International Conference on Pervasive Computing and Communication (PerCom'04*, pages 18–22, 2004.

[153] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[154] Nan Xu, Wei Shi Zhang, Hua Dong Yang, Xiu Guo Zhang, and Xing Xing. CACOnt: A Ontology-Based Model for Context Modeling and Reasoning. *Applied Mechanics and Materials*, 347–350:2304–2310, August 2013.

[155] Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer, and Gregor Engels. On the Development of Context-Aware Augmented Reality Applications. In José Abdelnour Nocera, Antigoni Parmaxi, Marco Winckler, Fernando Loizides, Carmelo Ardito, Ganesh Bhutkar, and Peter Dannenmann, editors, *Beyond Interactions*, volume 11930, pages 107–120. Springer International Publishing, Cham, 2020.

[156] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100:9–34, January 1999.

[157] Daqiang Zhang, Hongyu Huang, Chin-Feng Lai, Xuedong Liang, Qin Zou, and Minyi Guo. Survey on context-awareness in ubiquitous media. *Multimedia Tools and Applications*, 67(1):179–211, November 2013. context modeling and reasoning.

[158] J. Zhu, S.K. Ong, and A.Y.C. Nee. A context-aware augmented reality assisted maintenance system. *International Journal of Computer Integrated Manufacturing*, 28(2):213–225, February 2015.

Appendix A

Background information questionnaire

Assessment of the DARLENE Augmented Reality smart glasses for the support of policing tasks

Background information questionnaire

Please respond to the questions below regarding background information for yourself

1. What is your age?

○ Under 18
○ 18-24
○ 25-34
○ 35-44
○ 45-54
○ 55-64
○ 65-74

2. What is your gender?

○ Male
○ Female
○ Other
○ Prefer not to answer

3. Do you work as a Law Enforcement Agent?

○ Yes

○ No

4. How many years of professional expertise do you have?

○ Less than 5 years
○ 5 to 10 years
○ More than 10 years

5. Do you have any professional expertise in:

	No experience	Less than 5 years	5 to 10 years	More than 10 years
crisis management and healthcare provision after cyber-attack or terrorist incident	○	○	○	○
criminal / terrorist apprehension	○	○	○	○
explosives neutralization	○	○	○	○
hostage incident management	○	○	○	○

6. Do you have any previous (security or non-security related) experience with Augmented Reality (AR) applications?

○ Yes

○ No

Appendix B

Debriefing interview

Code:_____

Assessment of the DARLENE Augmented Reality smart glasses for the support of policing tasks

Debriefing interview queries

Overall, how would you describe your experience with the DARLENE system?

_____

_____

_____

_____

Did you feel nauseous while using the DARLENE Augmented Reality glasses?

YES NO

(If yes) On a scale of 1 to 10, how intense would you rate the nausea you felt?

1	2	3	4	5	6	7	8	9	10

Do you think that it can assist you in becoming more aware of a situation at hand you may be facing?

_____

_____

_____

_____

_____

Do you think that it will be difficult to use during field operations?

_____

_____

_____

_____

_____

What did you like most about the system?

Code:_____

_____

_____

_____

_____

What did you like least?

_____

_____

_____

_____

What would you like the system to have that it currently doesn't?

_____

_____

_____

_____

_____

Would you like to use it during operations or not, and why?

_____

_____

_____

_____

_____

Appendix C

SAGAT questionnaire

SAGAT Questionnaire

Video 1 (10 questions)

Part 1:

1. How many Victims are in the scene?
   a) b) c)
2. How many weapons is the man with the red t-shirt carrying?
   a) b) c)
3. Which of the following weapons is the man with the red t-shirt carrying?
   a) b) c)

Part 2:

1. How many Victims are bleeding?
   a) b) c)
2. Has anyone been shot?
   a) b)
3. Are there any policemen in the scene?
   a) b)
4. To whom the paramedics need to give priority?
   a) b) c)

Part 3:

1. How many threats remain to be neutralized?
   a) b) c)
2. Are there any unattended objects in the scene?
   a) b)
3. What does the red bag contain?
   a) b) c)

Video 2 (6 questions)

Part 1:

1. What weapon does the female perpetrator possess?
   a) b) c)
2. What weapon does the male perpetrator possess?
   a) b) c)
3. What weapon does the female perpetrator use?
   a) b) c)

Part 2:

1. How many paramedics are in the scene?
   a) b) c)
2. How many people need medical assistance?
   a) b) c)
3. Who do you expect to neutralize the threat?
   a) b) c)

Video 3 (9 questions)

Part 1:

1. How many Foes got out of the car?
   a) b) c)
2. How many of the perpetrators remain to be neutralized?
   a) b) c)
3. Did any Foe enter the building?
   a) b)
4. Has any policeman been shot?
   a) b)
5. How many policemen are in the scene?
   a) b) c)

Part 2:

1. What weapon does the perpetrator possess?
   a) b) c)
2. What weapon does the perpetrator use?
   a) b) c)
3. Which Victim is in a less critical condition?
   a) b) c)
4. How many of the perpetrators have been neutralized?
   a) b) c)

Video 4 (8 questions)

Part 1:

1. How many people are acting suspiciously?
   a) b) c)
2. Why is the person in the entrance acting suspiciously?
   a) b) c)

Part 2:

1. Are there any paramedics in the scene?
   a) b)
2. To which of the following people the paramedics need to give priority?
   a) b) c)
3. Who is less critically injured?
   a) b) c)
4. How many people need medical assistance?
   a) b) c)

Part 3:

1. How many policemen are in the scene?
   a) b) c)
2. How many Victims are in the scene?
   a) b) c)

Video 5 (10 questions)

Part 1:

1. How many are the perpetrators?
   a) b) c)
2. Which of the following weapons does this perpetrator possess?
   a) b) c)
3. Which of the following weapons does this perpetrator possess?
   a) b) c)
4. How many people has this Foe attacked?
   a) b) c)
5. Has anyone been shot?
   a) b)

Part 2:

1. How many perpetrators remain to be neutralized?
   a) b) c)
2. Which perpetrator has been neutralized?
   a) b)
3. Is this perpetrator fully disarmed?
   a) b)
4. How many Allies have neutralized the perpetrator?
   a) b) c)
5. Is any policeman wounded?
   a) b)

Video 6 (5 questions)

Part 1:

1. Are there any policemen in the scene?
   a) b)
2. Are there any suspicious people in the scene?
   a) b)
3. In what kind of incident will the situation unravel?
   a) b) c)

Part 2:

1. How many policemen are in the scene?
   a) b) c)
2. How many perpetrators remain to be neutralized?
   a) b) c)

Video 7 (5 questions)

Part 1:

1. How many are the perpetrators?
   a) b) c)
2. How many perpetrators have been neutralized?
   a) b) c)
3. What weapons are being used by the perpetrators?
   a) b) c)
4. In which parts of the ship are there perpetrators?
   a) b) c)

Part 2:

1. Are there any civilians in need of medical assistance?
   a) b)

Video 8 (5 questions)

Part 1:

1. Which of the following weapons are the perpetrators carrying?
   a) b) c)
2. How many civilians have been shot?
   a) b) c))
3. Which of the following people have been shot?
   a) b) c)
4. How many perpetrators have been neutralized?
   a) b) c)
5. Are there any policemen in the scene?
   a) b)