Received January 11, 2022, accepted February 12, 2022, date of publication February 18, 2022, date of current version March 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152743

Real-Time Adaptation of Context-Aware Intelligent User Interfaces, for Enhanced Situational Awareness

ZINOVIA STEFANIDI[1], GEORGE MARGETIS[1], STAVROULA NTOA[1], AND GEORGE PAPAGIANNAKIS[1,2], (Member, IEEE)

[1]Institute of Computer Science, Foundation for Research and Technology—Hellas (FORTH), 70013 Heraklion, Greece
[2]Department of Computer Science, University of Crete, 700 13 Heraklion, Greece

Corresponding author: George Margetis (gmarget@ics.forth.gr)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program through the Project "Deep AR Law Enforcement Ecosystem (DARLENE)" under Grant Agreement 883297.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Social and Societal Ethics Committee of the Katholieke Universiteit Leuven under KUL Approval No. G-2021 09 2072.

ABSTRACT In this work, a novel computational approach for the dynamic adaptation of User Interfaces (UIs) is proposed, which aims at enhancing the Situational Awareness (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 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). 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.

INDEX TERMS Adaptive user interfaces, augmented reality, context-awareness, intelligent user interfaces, ontology modeling, ontology reasoning, situational awareness, user interface optimization.

I. INTRODUCTION

User Interfaces (UIs) constitute the prominent means for interacting with computing systems and applications, with a decisive impact in their utility, accessibility and the overall User Experience (UX). 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. The concept of extracting information from the environment and reacting to the changing requirements of use has been coined in the literature as 'Context-Awareness' [1]. 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 [2]. 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

The associate editor coordinating the review of this manuscript and approving it for publication was Charalambos Poullis.

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 [3]. 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 [4]. 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 [5], or minimizing user effort [6]. 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 [4], are defined a priori and do not reflect the variable and dynamic context in which the goal of the UI needs to be optimized. 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) [7]. 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'* [8]. In particular, the theoretical model of SA [9] 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 incommode SA - the so called "SA demons" [10]. Prominent such factors include stress, anxiety and workload, taxing attention and working memory, as well as information overload, when data exceed the human capacity. 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, 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. The main goal of the proposed approach is to optimize the SA associated with the displayed UI *at run-time*, while avoiding "SA demons", such as information overload and induced stress.

The novelties 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* [11], which investigates means by which Augmented Reality (AR) and Machine Learning (ML) can be employed, in real time, to improve the SA of LEAs 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 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. 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 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 (objective) and perceived (subjective) user SA is improved, by 9.25% and 25.63% respectively.

II. RELATED WORK

This section carries out a review of related work, elaborating on topics relevant to our approach, and in particular: Ontology-based modeling and reasoning, employed to make use of the available context; context-aware adaptive UIs, supporting the adaptation of UIs based on the context; 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.

A. ONTOLOGY BASED MODELING AND REASONING

Context-Awareness was first introduced in the domain of ubiquitous computing, and has since rapidly expanded to other research areas, including Intelligent User Interfaces (IUIs) and 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 [12], [13].

An Ontology is a formal description of the concepts and relationships present in a given domain. 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),[1] the Resource Description Framework (RDF),[2] and the Resource Description Framework Schema (RDFS).[3] OWL is the prevalent one, being more expressive [14] and facilitating greater machine interpretability of Web content, through additional vocabulary as well as formal semantics.[4] Some notable ontology-based context models that have been proposed in the literature are the Context Broker Architecture (CoBrA) project [15], the Context Ontology (CONON) project [16], the Service-Oriented Context-Aware Middleware (SOCAM) architecture [17], the Context-Driven Adaptation of Mobile Services (CoDAMoS) project [18], the general and extensible context-aware computing ontology (CACOnt) [19], and the Context Awareness Meta Ontology modeling (CAMeOnto) [20] used by a reflective middleware for context-aware applications, called CARMiCLOC.

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

As indicated in [21], 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',[5] 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 of Ontology-based modeling is the support for Ontology-based reasoning, deriving new knowledge based on the existing contextual information modeled in an Ontology and identifying potential inconsistencies. This approach uses description logic and is implemented by Semantic web languages, such as the Semantic Web Rule Language (SWRL),[6] which represents rule-based first-order logic (FOL) inference rules, expressed in terms of predefined OWL context knowledge. Such reasoning tools that are widely used are FaCT,[7] Hermit[8] and Pellet.[9]

B. CONTEXT AWARE ADAPTIVE UIs

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

Regarding mobile and desktop applications, and web pages, context and context-awareness have been thoroughly investigated, though a profusion of reviews, e.g. [22], [23], reference architectures and frameworks e.g. [23]–[26], and adaptation techniques, e.g. [2], [26]–[28] 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.

Reference architectures or frameworks for adapting UIs of interactive systems include the CAMELEON-RT [24], a conceptual reference architecture for developing distributed, migratable and plastic UI's, the TriPlet [23], a conceptual framework for context-aware adaptation of UIs which consists of three core components: a Context-Aware Meta-model (CAMM), a Context-Aware Reference Framework (CARF) and a Context-aware Design Space (CADS), the CEDAR [25], 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 and the AUI-UXA [26] which proposes a framework in the form of an adaptive UI/UX authoring tool.

Research on AUIs is focused on the development of handcrafted rules and heuristics, whose creation is carried out either with the help of UX experts, or system designers [26]. Furthermore, most of adaptive UI systems use Ontology models, for the purpose of storing the information for tailoring

[5]https://protege.stanford.edu/
[6]https://www.w3.org/Submission/SWRL/
[7]http://owl.cs.manchester.ac.uk/tools/fact/
[8]http://www.hermit-reasoner.com/
[9] https://www.w3.org/2001/sw/wiki/Pellet

the UI. For example, in the method and set of tools presented in [27], end users without programming experience can customize the application UI and/or logic, using trigger-action rules. Another similar approach is the ISATINE framework [28] that proposes 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 work in [2], presents an ontology-based approach for automatically suggesting adaptive UIs according to the context of use, using SWRL rules. Apart from these rule-based or heuristic approaches, Combinatorial Optimization (CO) 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.

In their majority, existing approaches focus 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 Mixed Reality applications, calling for more dynamic approaches. Lindlbauer *et al.* [5] 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 IUIs, 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 the parameters of the optimization problem are reasoned at run-time from the context model, instead of being static or hard-coded. This applies both for the constraints and the objective function, contrary to the state-of-the-art.

C. COMBINATORIAL OPTIMIZATION FOR UI GENERATION AND ADAPTATION

An emerging technique for the automatic generation and adaptation of UIs, is Combinatorial Optimization (CO), as surveyed in the articles by Oulasvirta *et al.* [4], [29]. It is a powerful and flexible tool for formulating interface adaptation or generation as an optimization problem, defining constraints and maximizing (or minimizing) an objective function that represents the goal of the UI, for instance maximizing the interface's usability [5], or minimizing user effort or selection time e.g. [6].

Our approach uses CO for adaptation and personalization, modifying the UI at run-time based on the user and the current context of use. A first approach towards this direction was SUPPLE [30], 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 [31] focuses on ability-based optimization, where UIs are adapted by considering the user's motor or cognitive impairments. SUPPLE++ [32] 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.* [33] explores a computational design approach using CO for improving UI designs for users with sensorimotor and cognitive impairments.

Based on the categorization of UI design CO problems in [4], the classes relevant to our approach are selection and layout problems. Selection problems 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 0-1 knapsack problem. Layout problems involve fitting a set of given objects onto a canvas, while satisfying feasibility constraints, such that there is no overlap or overflow. Our optimization problem combines characteristics of both selection and layout problems. Similar to the knapsack problem, it tries to select the Component Types and Levels of Detail (LoDs) 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, 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 Components, without overlapping with others. These selection and layout dimensions are being solved simultaneously, the one affecting the outcome of the other, utilizing better the available display space and adjusting the presented content based on positional constraints. This is contrary to current approaches, which separate the decision of what virtual components to place and how, with the problem of where to place them. As an example, in the work by Lindlbauer *et al.* [5], it is first determined by the optimization step which UI elements are displayed and at which level of detail (LoD) and then, as a final step, their placement is specified.

D. CONTEXT-AWARE MIXED REALITY

In general, there exist different frameworks, e.g. [34], [35], and applications, e.g. [5], [36], targeting the area of context-aware Mixed Reality. 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.* [37] 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.* [36]

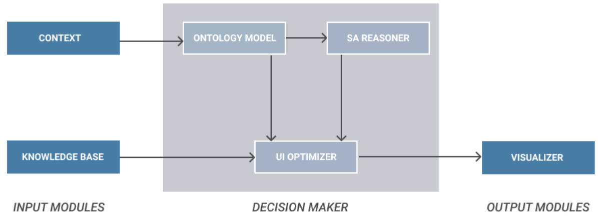

FIGURE 1. The interface modules and the units of the Decision Maker (DM).

proposed adapting the displayed augmentation based on the scene illumination, the distance to the target and the ambient noise, using appropriate formulas. Contrary to previous approaches, Lindlbauer *et al.* [5] formulates a CO problem whose solution adapts the virtual content and its presentation based on the users' cognitive load and task. Similar to [5], 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 [38]. Our system, similarly to the works presented in [5], [37], adopts this concept using different LoD presentations of content based on the current context.

III. 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 SA of the user. The Decision Maker (DM) 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:

- 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 *DARLENE*, Deep Neural Networks (DNNs) were constructed, to detect user stress, as presented in [39].
- The Knowledge Base module (KB), which provides to the DM the necessary data (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 work, 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 III-B, III-C and III-D respectively.

The interface modules and the units of the DM are portrayed in Figure 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

TABLE 1. List of component types and their description.

Component Type	Description
Suspect Detection	Highlights suspects (e.g. persons behaving oddly) in the LEA's field of view
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)
Alerts	Gives urgent information from the Command & Control (e.g. arrest John Doe)

TABLE 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
Task	The task the LEA is currently executing
Device	The device the LEA is currently utilizing

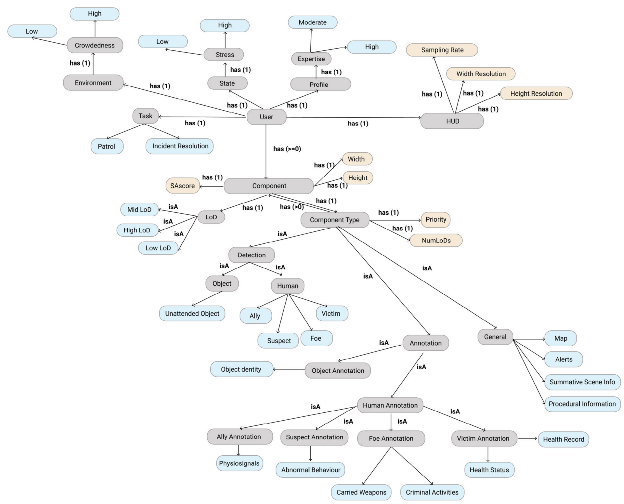

FIGURE 2. DARLENE ontology model.

procedure for the target application domain, described in section III-A and the development of the DM units in sections III-B, III-C and III-D.

A. 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 [40]. In these workshops, a total of 30 end users (LEAs) participated, from police agencies in 5 countries.

The systematic analysis of the outcomes of there workshops, which were analyzed following a combination of deductive and inductive coding, 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, as presented in section III-B. Three examples of the currently supported Component Types and their description are presented in Table 1. In addition, context factors currently considered are presented in Table 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.

B. ONTOLOGY MODEL

In the Ontology Model unit of the DM, we model the studied application domain, based on the user requirements. For the definition of the Ontology, relevant context factors are captured, following a similar approach to [41]. Furthermore, all the supported GUI elements for display, along with accompanying metadata (e.g. their dimensions) are also represented. Regarding the *DARLENE* use case, the graphical representation of the Ontology is depicted in Figure 2. It is logically separated into two parts, the context factors, and the GUI elements, explained in the subsequent subsections III-B1 and III-B2, respectively.

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, as described in section III-D.

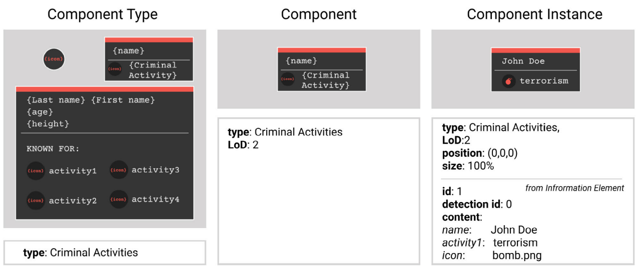

FIGURE 3. An example of the properties corresponding to a component type, component and component instance, as well as their relation.

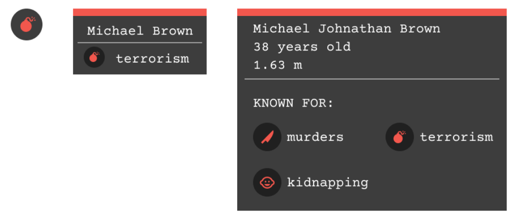

FIGURE 4. An example of the 3 LoDs of the component type 'Criminal Activities', from low to high, instantiated with context, as displayed by the visualizer.

2) GUI ELEMENTS

The second logical part of the Ontology represents in a hierarchical manner 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 (examples in Table 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, along with their corresponding Information Element, do not need to be included in the Ontology, as explained later in this section. In figure 3, we can see the related information and properties for the notions of Component Type, Component and Component Instance, through an example.

a: 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 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 III-C.

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 4 an example of the 3 LoDs of the Component Type 'Criminal Activities' is depicted, instantiated with content (Component Instances), as displayed 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.

b: 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. 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. Besides information regarding their size, another modeled data property of Components, perhaps the most significant, is the 'hasSAscore'.

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 III-C. 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 III-D.

c: 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 4. 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 and corresponding Information Elements 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) ONTOLOGY DEFINITION

The Ontology was built in the OWL 2.0 Web Ontology Language. For the purpose of designing and visualizing our Ontology, the software 'Protégé'[10] 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 III-C, Owlready2,[11] a package for ontology-oriented programming in Python was utilized.

C. SA REASONER

In the SA Reasoner unit of the DM, an SA score for each Component in the Ontology of III-B 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 the Stress level. Each time the context changes, the SA Reasoner recalculates the SA scores and propagates them to the UI Optimizer, described in section III-D.

1) 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 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 the LEA application domain, which was studied, 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.

2) SWRL RULES DEFINITION

For the *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'[12] OWL 2 reasoner was employed for this purpose. These rules can be divided into

[10] https://protege.stanford.edu/
[11] https://pypi.org/project/Owlready2/
[12] https://www.w3.org/2001/sw/wiki/Pellet

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.

a: 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, Where for each Task and Component Type, the corresponding Priority is set:

User(?u), Component(?c), hasTask(?u,**{Task}**), hasComponent(?u,?c), hasComponentType(?c, **{ComponentType}**)-> hasPriority(**{ComponentType}**, **{Priority}**)

b: 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 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.

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. In particular, 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:

$$1 - 0.1 \cdot Priority \tag{1}$$

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. The rules, encompassing this method for computing the SA score, 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) -> hasSAscore(?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. 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,**{LoD}**), hasPriority(?cT,?r), multiply(?s1,−0.1,?r), add(?s2,1,?s1), add(?s,**{LoDSA}**,?s2) -> (?c,?s)

Where for HighStress and each possible value of Crowdedness, and LoD, the LoDSA, which is the appropriate value

FIGURE 5. A display grid with sampling rate 60px.


```
{
    "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 6. 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').

from the set 0.09, 0.05, 0.01, is added to (1) to produce the SA.

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.

D. 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.

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 5. 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 1px, 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 III-D2.

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.

- 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.

TABLE 3. 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}

- 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 information regarding the current scene, which 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. In Figure 6, 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) OPTIMIZATION PARAMETERS

Given these input sources, the parameters of the optimization problem can be defined (Table 3).

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.

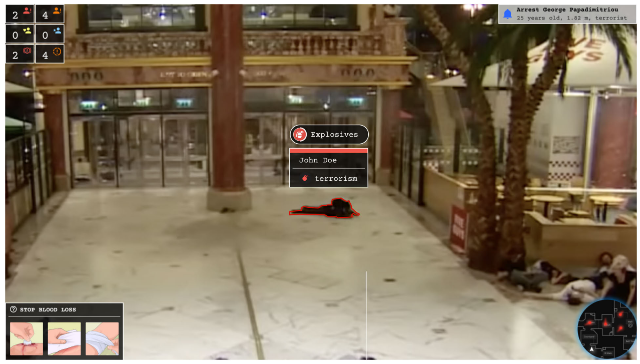

FIGURE 7. An example of possible positions for 'Annotation' and 'General' components. The 'General' components are placed relative to the display, e.g. on the four corners, whereas the 'Annotation' components are placed relative to the corresponding detection.

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.

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 7. 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 7. 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 [42].

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 corresponding Component Type t (thus $|E| = |T|$). 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:

$$f(x) = \sum_{t} \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} * (a_{c_t} + y_{p_{c_t}}) \tag{2}$$

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

$$\max_{x} f(x) \; subject\ to\ v(x),\ c(x),\ u_t(x) \quad \forall t \in \{1, \ldots, n\} \tag{3}$$

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:

a: 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:

$$u_t(x) = \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} \leq 1, \quad \forall t \in \{1, \ldots, n\} \tag{4}$$

b: 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:

$$v(x) = \sum_{t} \sum_{c_t} \sum_{p_{c_t}} x_{p_{c_t}} \leq N \tag{5}$$

c: 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:

$$\begin{aligned} &\forall p1_{c1_{t1}}, p2_{c2_{t2}} \\ &\qquad x_{p1_{c1_{t1}}} * x_{p2_{c2_{t2}}} * isCollided(p1_{c1_{t1}}, p2_{c2_{t2}}) = 0, \\ &\qquad p1_{c1_{t1}} \neq p2_{c2_{t2}} \end{aligned} \tag{6}$$

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.

d: 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 $\frac{\mathit{Width\ Resolution}}{\mathit{Sampling\ Rate}} \times \frac{\mathit{Height\ Resolution}}{\mathit{Sampling\ Rate}}$ 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*.

e: 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.

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 8, 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

FIGURE 8. Positioning of an annotation in terms of grid tiles (left) and in terms of pixels (right). The sampling rate is 60px.

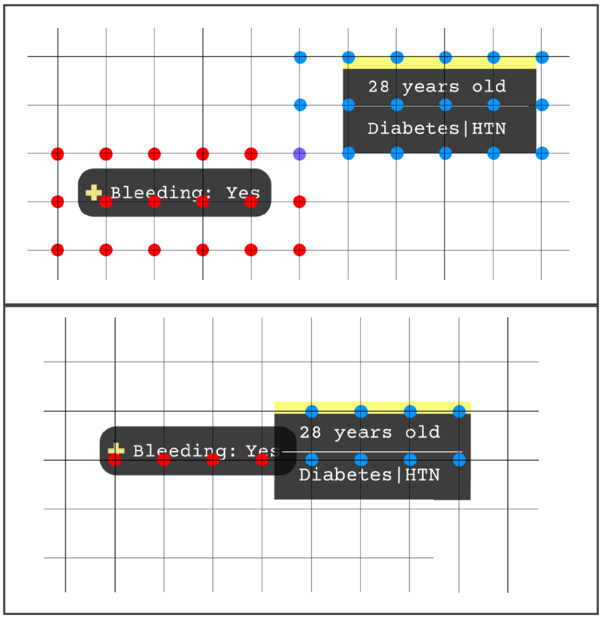

FIGURE 9. An example of the ''over-sensitive'' (top) and ''under-sensitive'' (bottom) coarse collision approach.

is smaller than the smallest Component. In Figure 9 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 section V, 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.

6) SOLVING THE OPTIMIZATION PROBLEM

For the purpose of solving the integer linear program defined in equations (2) – (6), the *Gurobi*[13] mathematical optimization solver is used. An initial implementation with the *CPLEX*[14] 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 section IV, the DM was run with display resolution 1920×1080 and Sampling rate 60px. The number of LoDs for each Component were 3 and the 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 section V, 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.

[13]https://www.gurobi.com/
[14]https://www.ibm.com/analytics/cplex-optimizer

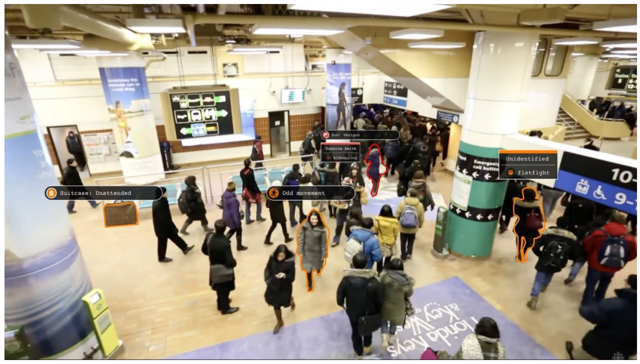

FIGURE 10. The generated UI for Scenario B: Patrolling - high crowdedness.

IV. EXPERT BASED EVALUATION

As a first qualitative assessment of our computational approach, an expert-based evaluation was conducted, involving 10 experts: 5 User Experience (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. Such evaluations are suggested to be used early in the development life-cycle, and in complementarity with user testing [43]. 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 various experimental conditions. For each experimental condition, our algorithm was run on selected 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.

A. 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). 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 Figure 10, 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 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 alleviate carryover effects [44].

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 [45] 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 response, 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.

B. RESULTS

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.

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.

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.

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.

C. 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 non-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.

D. 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.

V. USER BASED EVALUATION

To explore further benefits and limitations of our improved computational approach and investigate whether it leads to an enhanced Situational Awareness 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 [46], [47]. 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 [48], 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

FIGURE 11. A frame of one of the simulation videos, in a non-stress condition.

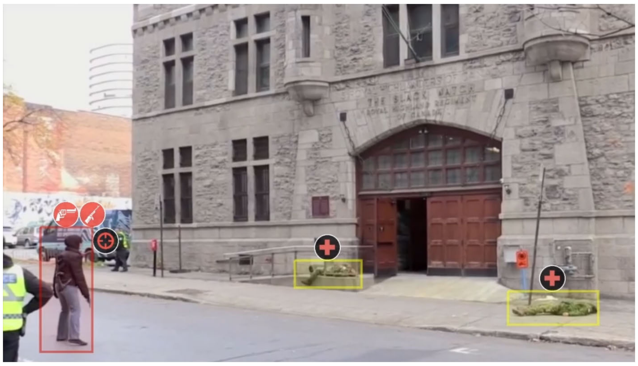

FIGURE 12. A frame of one of the simulation videos, in a stress condition.

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 11 and 12, frames from two different simulation videos are displayed, in non-stressful and stressful conditions, respectively.

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

A. METHODOLOGY

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

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 *non-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 4 × 4 Latin square design and assigning the simulated scenarios to the following conditions *with System-Stress*, *without System-Stress*, *with System-non-Stress*, and *without System-non-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 [49] was administered for estimating the perceived SA, whereas the SAGAT query technique [48] was employed for the observed SA. For measuring the workload and overall UX, the NASA-TLX [50] and UMUX-Lite [51] questionnaires were completed. All questionnaires employed in this study are standardized, ensuring the validity and accuracy of results.

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 [48]. 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 [52]. The SAGAT questions for our application domain were developed in line with the SA requirements highlighted in the requirements elicitation and analysis procedure, 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 [9].

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 [53]. 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 [54]. In our experiment, we employed the Paced Auditory Serial Addition Test (PASAT) [55], 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 [56].

4) APPARATUS AND VIDEOS

To simulate the experience of policing using AR glasses, the VR Google Cardboard Headset was utilized. The simulation

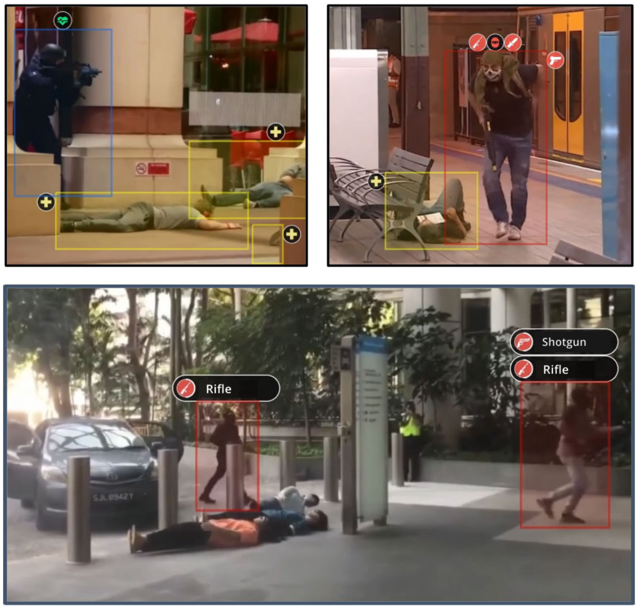

FIGURE 13. **Collage of frames from different simulation videos.**

videos were streamed on a Google Pixel 5 Android Phone, placed inside the headset. They comprised of 8 short (< 1 min) 3D stereoscopic 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 13, frames from different simulation videos are displayed, augmented with GUI elements.

B. PROCEDURE

The experiment was structured in three phases: introduction, main part of the study, and debriefing Participants were first welcomed and explained the aim and objectives of the study.[15] 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 video featuring nature images and relaxing music for 5 minutes, to prepare for the *non-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.

C. 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. The vast majority of the participants (85%) did not have prior experience with AR.

D. 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 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.

1) SITUATIONAL AWARENESS

In order to study the Situational Awareness 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* (AD), *Attentional Supply* (AS) and *Understanding* (U). The score for each subscale is

[15]This study has been approved by the Social and Societal Ethics Committee of the Katholieke Universiteit Leuven (KUL approval number G-2021 09 2072)

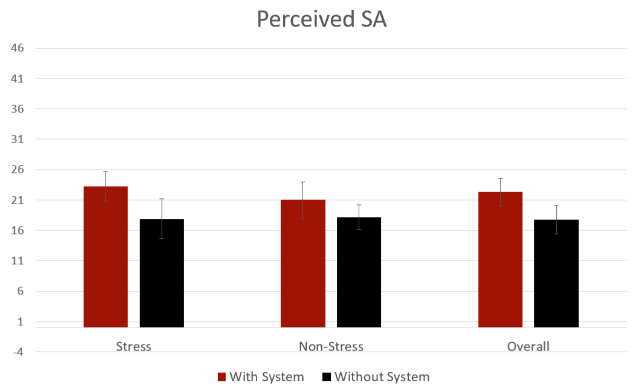

FIGURE 14. Perceived SA, in both stress states and overall.

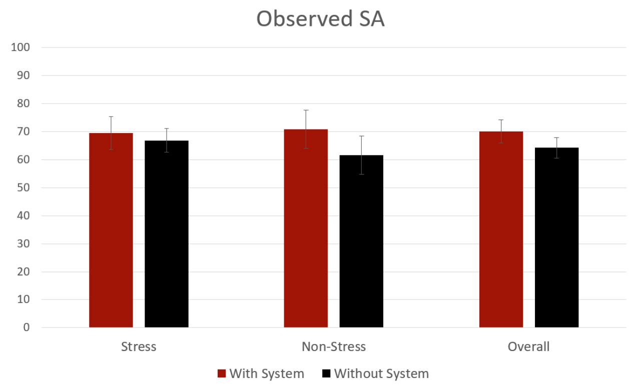

FIGURE 15. Observed SA, in both stress states and overall.

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.

$$SARTscore = U - (AD - AS) \tag{7}$$

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 14 for the perceived SA and in Figure 15 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*.

a: PERCEIVED SA

The overall perceived SA was enhanced when using the system, both in the stress (*hypothesis H1a*) and non-stress condition (*hypothesis H1b*).

In the stress condition, with the system, the SART score was above the midpoint (16) of its range ([−14, 46]) for all participants, reflecting that they had good SA during the simulation scenarios corresponding to this condition. On the contrary, when not using the system, 30% of the participants had a SART score below the midpoint of this range, highlighting that some of them perhaps felt that their SA was not so good. In more detail, with respect to the different subscales, *Attentional Demand* was perceived higher when not using the system, whereas *Attentional Supply* and *Understanding* were perceived higher with the system. In Figure 16, we can observe the differences in the average scores for the individual SART subscales and the final SART score, with the system and without the system, for the stress condition.

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 ($M = 17.89$, $SD = 6.87$); $t(18) = 2.44$, $p = 0.025$.

In the non-stress condition, with 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. 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. In Figure 17, we can observe the differences in the average scores for the individual SART subscales and the final SART score, with the system and without the system, for the non-stress condition.

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.

b: OBSERVED SA

Regarding the observed SA, the performance of the participants improved when using the system, both in the stress (*hypothesis H1a*) and non-stress condition (*hypothesis H1b*).

In the stress condition, users of the system achieved a better SA score on average ($M = 69.52$, $SD = 12.66$), compared to participants without the system ($M = 66.88$, $SD = 8.96$). 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

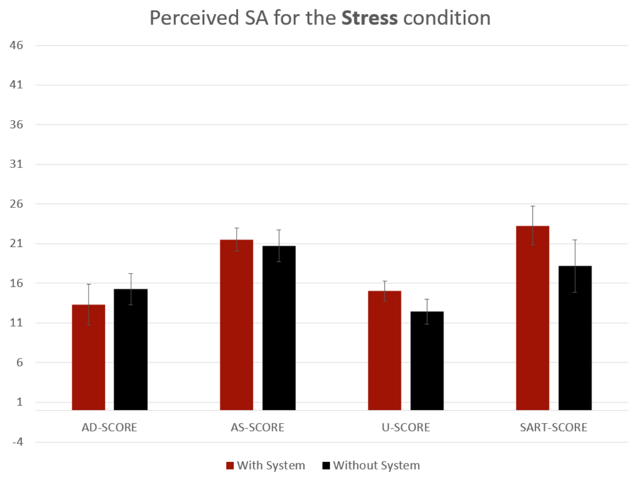

FIGURE 16. The average participants' scores for the individual SART subscales and the resulting SART score, with and without the system, in the stress condition.

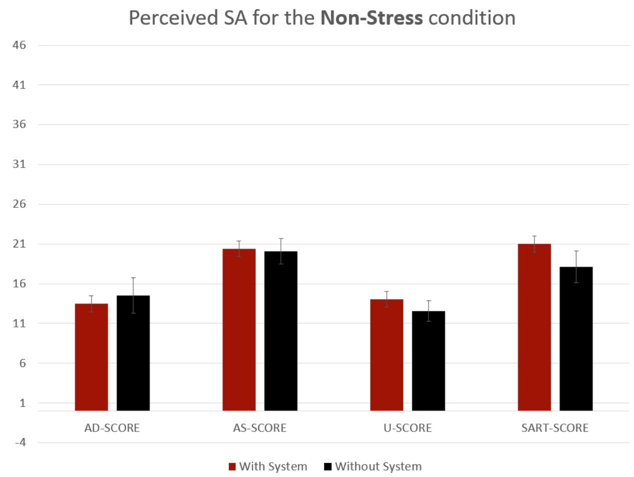

FIGURE 17. The average participants' scores for the individual SART subscales and the resulting SART score, with and without the system, in the non-stress condition.

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.

In the non-stress condition, 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 ($M = 61.62$, $SD = 14.49$); $t(19) = 2.24$, $p = 0.03$.

c: 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.

d: CONCLUSION

In conclusion, the Situational Awareness of participants was improved when using the system both in the *Stress Condition* and the *Non-stress Condition*, confirming our hypothesis (*H1a* and *H1b*). In the *Stress Condition*, their perceived SA exhibited statistically important differences with the system, in comparison to without it, whereas in the *Non-stress Condition*, differences were statistically important regarding the observed SA. Furthermore, these findings were supported by qualitative feedback, provided during the debriefing session.

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.

Regarding the workload of the participants in stress conditions while using the system (hypothesis *H2a*), the average of the overall workload score was **58.33**. With respect to the different dimensions of the NASA-TLX, 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.

With regard to 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**, which 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.

In order to better understand these scores, in Figure 18 we summarize these results in comparison to findings from a study with police officers in a field shooting exercise [57]. 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 non-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).

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. Moreover, the qualitative feedback obtained during the debriefing session was considered.

a: UMUX-LITE RESULTS

The overall UX score as well as the score for each dimension of the UMUX-Lite can be observed, for both the stress and non-stress conditions, in Figure 19.

In the stress condition, the overall UX score across participants was 5.03 out of 7 (stdev: 1.50), with 95%CIs

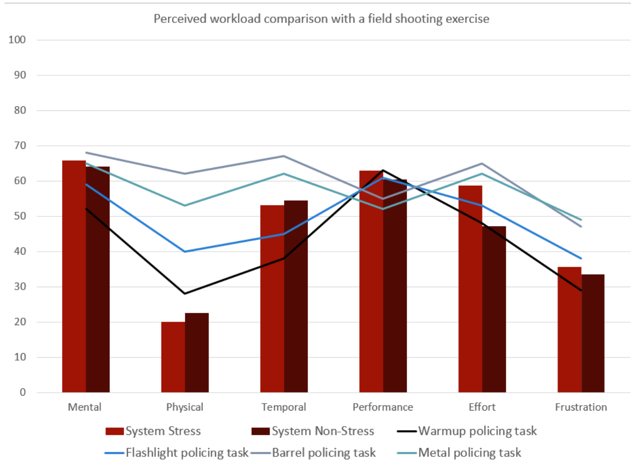

FIGURE 18. Perceived workload comparison with a field shooting exercise.

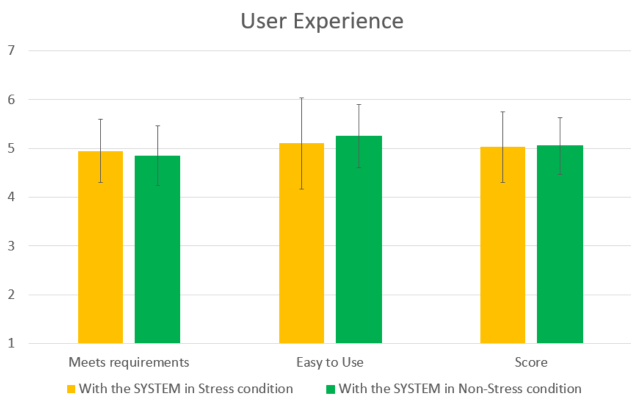

FIGURE 19. The results of the UMUX-Lite questionnaire. The error bars represent the 95%CI.

[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).

Furthermore, 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 'usefulness' question 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 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).

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.

b: 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

4) FURTHER TESTS

Apart from our initial, core research questions, we also assessed the impact of stress and expertise in the perceived and observed SA, the workload, and UX of the participants, both with and without using the system. Moreover, we examined for all SA levels of observed SA, the impact of using the system, as well as the effect of stress and expertise. Finally, we performed a correlation analysis of all the measurements acquired in this study.

a: IMPACT OF STRESS

To assess the effect of stress, paired two-tailed t-tests were carried out on the participants' scores regarding the perceived and observed SA, the workload, and UX, between the stress and non-stress condition.

When analysing the participants' SART scores when using the system, a statistically significant difference was found between the perceived SA when 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 non-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.

In addition, the analysis of the NASA-TLX effort score indicated 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.

The rest of the measures, yielded no significant effect between the stress and non-stress condition.

b: IMPACT OF EXPERTISE

To assess the impact of expertise, an expertise score for each individual was calculated as follows. For each question on the background information questionnaire, regarding professional expertise in different domains, 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.

When analysing the 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).

Regarding the 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.

Furthermore, 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.

However, 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 ($F(19.19) = 0.7468$, $p = 0.73$). Similarly, no significant effect was found for perceived SA when using the system ($F(18.18) = 0.5840$, $p = 0.86$). 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.

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.

c: SA LEVELS

As already mentioned, the theoretical model of SA [9] 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:

Regarding 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 ($F(19, 19) = 1.36$, $p = 0.25$).

Moreover, for Level 2 SA, in the no stress condition, the paired t-test conducted to compare results with the system (M = 68.49, SD = 15.07) and without it (M = 57.71, SD = 17.55), yielded 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

TABLE 4. Correlations between the measurements for the stress condition.

	UMUX-Lite	NASA-TLX	SART	SAGAT
UMUX-Lite	1			
NASA-TLX	0.14	1		
SART	0.37	0.26	1	
SAGAT	-0.31	0.03	-0.1	1

TABLE 5. Correlations between the measurements for the non-stress condition.

	UMUX-Lite	NASA-TLX	SART	SAGAT
UMUX-Lite	1			
NASA-TLX	-0.06	1		
SART	0.29	0.44	1	
SAGAT	-0.15	0.09	-0.07	1

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 ($F(19, 19) = 1.16$, $p = 0.38$).

With respect to the rest of the SA Levels, in both stress and non-stress conditions, no statistically significant difference was found when using the system.

Furthermore, paired t-tests were carried out to also explore the effect of stress on observed SA for the three different SA Levels. This resulted to a statistically significant difference 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, without the system; $t(19) = 2.41$, $p = 0.02$. 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. No statistically significant difference was found for the rest of the tests.

d: CORRELATIONS

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, as can be seen in Tables 4 and 5, respectively.

VI. CONCLUSION AND FUTURE WORK

In this work, we introduced a novel computational methodology, which aims at enhancing the Situational Awareness 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 demons", such as information overload and induced stress.

The proposed, general-purpose methodology was deployed 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 non-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.

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.

REFERENCES

[1] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proc. 1st Workshop Mobile Comput. Syst. Appl.*, Dec. 1994, pp. 85–90.

[2] M. Soui, S. Diab, A. Ouni, A. Essayeh, and M. Abed, "An ontology-based approach for user interface adaptation," in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2017, pp. 199–215.

[3] J. Hussain, A. U. Hassan, H. S. M. Bilal, R. Ali, M. Afzal, S. Hussain, and S. Lee, "Model-based adaptive user interface based on context and user experience evaluation," *J. Multimodal User Interfaces*, vol. 12, no. 1, pp. 1–16, 2018.

[4] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John, and A. Karrenbauer, "Combinatorial optimization of graphical user interface designs," *Proc. IEEE*, vol. 108, no. 3, pp. 434–464, Mar. 2020.

[5] D. Lindlbauer, A. M. Feit, and O. Hilliges, "Context-aware online adaptation of mixed reality interfaces," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, New Orleans, LA, USA, Oct. 2019, pp. 147–160.

[6] G. Bailly, A. Oulasvirta, T. Kötzing, and S. Hoppe, "MenuOptimizer: Interactive optimization of menu systems," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, St. Andrews, U.K., Oct. 2013, pp. 331–342.

[7] M. R. Endsley, B. Bolte, and D. G. Jones, *Designing for Situation Awareness: An Approach to User-Centered Design*. Boca Raton, FL, USA: CRC Press, 2003.

[8] M. R. Endsley, "Design and evaluation for situation awareness enhancement," *Proc. Hum. Factors Soc. Annu. Meeting*, vol. 32, no. 2, pp. 97–101, Oct. 1988.

[9] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 37, no. 1, pp. 32–64, Mar. 1995.

[10] R. M. Endsley, "SA demons: The enemies of situation awareness," in *Designing for Situation Awareness*. Boca Raton, FL, USA: CRC Press, 2016, pp. 50–61.

[11] K. C. Apostolakis, N. Dimitriou, G. Margetis, S. Ntoa, D. Tzovaras, and C. Stephanidis, "DARLENE—Improving situational awareness of European law enforcement agents through a combination of augmented reality and artificial intelligence solutions," *Open Res. Eur.*, vol. 1, no. 87, p. 87, 2021.

[12] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *Proc. Workshop*, 2004.

[13] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive Mobile Comput.*, vol. 6, no. 2, pp. 161–180, 2010.

[14] O. Curé and G. Blin, "Chapter three-RDF and the semantic web stack," RDF Database Syst., Morgan Kaufmann, Boston, MA, USA, Tech. Rep., 2015, pp. 41–80.

[15] H. Chen, T. Finin, and A. Joshi, "Using OWL in a pervasive computing broker," in *Proc. Workshop Ontologies Agent Syst. (AAMAS)*, Jul. 2003, pp. 9–16.

[16] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Proc. IEEE Annu. Conf. Pervasive Comput. Commun. Workshops (PerCom)*, Mar. 2004, pp. 18–22.

[17] T. Gu, X. Wang, and D. Zhang, "An ontology-based context model in intelligent environments," in *Proc. CNDS*, Jan. 2004, pp. 270–275.

[18] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere, "Towards an extensible context ontology for ambient intelligence," in *Proc. Eur. Symp. Ambient Intell.* Berlin, Germany: Springer, 2004, pp. 148–159.

[19] N. Xu, W. S. Zhang, H. D. Yang, X. G. Zhang, and X. Xing, "CACOnt: A ontology-based model for context modeling and reasoning," *Appl. Mech. Mater.*, vols. 347–350, pp. 2304–2310, Aug. 2013.

[20] J. Aguilar, M. Jerez, and T. Rodríguez, "CAMeOnto: Context awareness meta ontology modeling," *Appl. Comput. Informat.*, vol. 14, no. 2, pp. 202–213, Jul. 2018.

[21] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive Mobile Comput.*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[22] J. M. Gómez and T. Tran, "A survey on approaches to adaptation on the web," in *Emerging Topics and Technologies in Information Systems*. Hershey, PA, USA: IGI Global, 2009, pp. 136–152.

[23] V. G. Motti and J. Vanderdonckt, "A computational framework for context-aware adaptation of user interfaces," in *Proc. IEEE 7th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, Paris, France, May 2013, pp. 1–12.

[24] L. Balme, A. Demeure, N. Barralon, J. Coutaz, and G. Calvary, "CAMELEON-RT: A software architecture reference model for distributed, migratable, and plastic user interfaces," in *Ambient Intelligence* (Lecture Notes in Computer Science), P. Markopoulos, B. Eggen, E. Aarts, and J. L. Crowley, Eds. Berlin, Germany: Springer, 2004, pp. 291–302.

[25] P. A. Akiki, A. K. Bandara, and Y. Yu, "Engineering adaptive model-driven user interfaces," *IEEE Trans. Softw. Eng.*, vol. 42, no. 12, pp. 1118–1147, Dec. 2016.
[26] J. Hussain, A. U. Hassan, H. S. M. Bilal, R. Ali, M. Afzal, S. Hussain, J. Bang, O. Banos, and S. Lee, "Model-based adaptive user interface based on context and user experience evaluation," *J. Multimodal User Interfaces*, vol. 12, no. 1, pp. 1–16, Mar. 2018.
[27] G. Ghiani, M. Manca, F. Paternò, and C. Santoro, "Personalization of context-dependent applications through trigger-action rules," *ACM Trans. Comput.-Hum. Interact.*, vol. 24, no. 2, pp. 14:1–14:33, Apr. 2017.
[28] V. López-Jaquero, J. Vanderdonckt, F. Montero, and P. González, "Towards an extended model of user interface adaptation: The isatine framework," in *Engineering Interactive Systems*, J. Gulliksen, M. B. Harning, P. Palanque, G. C. van der Veer, and J. Wesson, Eds. Berlin, Germany: Springer, 2008, pp. 374–392.
[29] A. Oulasvirta, "User interface design with combinatorial optimization," *Computer*, vol. 50, no. 1, pp. 40–47, Jan. 2017.
[30] K. Gajos and D. S. Weld, "SUPPLE: Automatically generating user interfaces," in *Proc. 9th Int. Conf. Intell. User Interface (IUI)*, 2004, pp. 93–100.
[31] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Automatically generating user interfaces adapted to users' motor and vision capabilities," in *Proc. 20th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, 2007, pp. 231–240.
[32] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces," in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, 2008, pp. 1257–1266.
[33] S. Sarcar, J. P. P. Jokinen, A. Oulasvirta, Z. Wang, C. Silpasuwanchai, and X. Ren, "Ability-based optimization of touchscreen interactions," *IEEE Pervasive Comput.*, vol. 17, no. 1, pp. 15–26, Jan. 2018.
[34] S. Krings, E. Yigitbas, I. Jovanovikj, S. Sauer, and G. Engels, "Development framework for context-aware augmented reality applications," in *Proc. 12th ACM SIGCHI Symp. Eng. Interact. Comput. Syst.*, Sophia Antipolis, France, Jun. 2020, pp. 1–6.
[35] E. Yigitbas, I. Jovanovikj, S. Sauer, and G. Engels, "On the development of context-aware augmented reality applications," in *Beyond Interactions*, vol. 11930, J. A. Nocera, A. Parmaxi, M. Winckler, F. Loizides, C. Ardito, G. Bhutkar, and P. Dannenmann, Eds. Cham, Switzerland: Springer, 2020, pp. 107–120.
[36] N. Ghouaiel, J.-M. Cieutat, and J.-P. Jessel, "Adaptive augmented reality: Plasticity of augmentations," in *Proc. Virtual Reality Int. Conf. (VRIC)*, New York, NY, USA, Apr. 2014, pp. 1–4.
[37] J. Zhu, S.-K. Ong, and A. Y. C. Nee, "A context-aware augmented reality assisted maintenance system," *Int. J. Comput. Integr. Manuf.*, vol. 28, no. 2, pp. 213–225, Feb. 2015.
[38] S. DiVerdi, T. Hollerer, and R. Schreyer, "Level of detail interfaces," in *Proc. 3rd IEEE ACM Int. Symp. Mixed Augmented Reality*, Arlington, VA, USA, Nov. 2004, pp. 300–301.
[39] K. Tzevelekakis, Z. Stefanidi, and G. Margetis, "Real-time stress level feedback from raw ecg signals for personalised, context-aware applications using lightweight convolutional neural network architectures," *Sensors*, vol. 21, no. 23, p. 7802, Nov. 2021.
[40] R. B. Rill and M. M. Hämäläinen, *The Art of Co-Creation: A Guidebook for Practitioners*. Singapore: Palgrave Macmillan, Aug. 2018.
[41] G. Margetis, S. Ntoa, M. Antona, and C. Stephanidis, "Augmenting natural interaction with physical paper in ambient intelligence environments," *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 13387–13433, May 2019.
[42] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.
[43] T. Hollingsed and D. G. Novick, "Usability inspection methods after 15 years of research and practice," in *Proc. 25th Annu. ACM Int. Conf. Design Commun. (SIGDOC)*, New York, NY, USA, Oct. 2007, pp. 249–255.
[44] L. Joseph Brooks, "Counterbalancing for serial order carryover effects in experimental condition orders," *Psychol. Methods*, vol. 17, no. 4, pp. 600–614, Dec. 2012.
[45] World Leaders in Research-Based User Experience. *Success Rate: The Simplest Usability Metric*. Accessed: Jan. 6, 2022. [Online]. Available: https://www.nngroup.com/articles/successrate-the-simplest-usability-metric/
[46] J. D. Winter, P. M. van Leeuwen, and R. Happee, "Advantages and disadvantages of driving simulators: A discussion," in *Proc. Measuring Behav.*, 2012, p. 8.
[47] V. Forsman, "Measuring situation awareness in mixed reality simulations," M.S. thesis, Dept. Comput. Sci., Mälardalen Univ., Västerås, Sweden, 2019.
[48] M. R. Endsley, "Situation awareness global assessment technique (SAGAT)," in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, Dayton, OH, USA, May 1988, pp. 789–795.
[49] J. S. Selcon and R. M. Taylor, "Evaluation of the situational awareness rating technique (SART) as a tool for aircrew systems design," AGARD, Situational Awareness Aerosp. Oper., NATO Sci. Technol. Org., France, Tech. Rep., 1990, p. 8.
[50] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," in *Advances in Psychology*, vol. 52, P. A. Hancock and N. Meshkati, Eds. North-Holland, The Netherlands: Elsevier, Jan. 1988, pp. 139–183.
[51] J. R. Lewis, B. S. Utesch, and D. E. Maher, "UMUXLITE: When there's no time for the SUS," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, Apr. 2013, pp. 2099–2102.
[52] M. R. Endsley, "Direct measurement of situation awareness: Validity and use of SAGAT," in *Situational Awareness*. Evanston, IL, USA: Routledge, 2017, pp. 129–156.
[53] A. Bali and A. S. Jaggi, "Clinical experimental stress studies: Methods and assessment," *Rev. Neurosci.*, vol. 26, no. 5, pp. 555–579, Oct. 2015.
[54] G. Giannakakis, D. Grigoriadis, K. Giannakaki, O. Simantiraki, A. Roniotis, and M. Tsiknakis, "Review on psychological stress detection using biosignals," *IEEE Trans. Affect. Comput.*, early access, Jul. 9, 2019, doi: 10.1109/TAFFC.2019.2927337.
[55] T. Tombaugh, "A comprehensive review of the paced auditory serial addition test (PASAT)," *Arch. Clin. Neuropsychol.*, vol. 21, no. 1, pp. 53–76, Jan. 2006.
[56] S. T. Mueller and B. J. Piper, "The psychology experiment building language (PEBL) and PEBL test battery," *J. Neurosci. Methods*, vol. 222, pp. 250–259, Jan. 2014.
[57] T. Oron-Gilad, J. Szalma, S. Stafford, and P. Hancock, "The workload and performance relationship in the real world: A study of police officers in a field shooting exercise," *Int. J. Occupational Saf. Ergonom.*, vol. 14, pp. 31–119, Feb. 2008.

ZINOVIA STEFANIDI holds a Bachelor and a Master's degree in Computer Science and Engineering from the Department of Computer Science of the University of Crete, Greece. During her studies, she has been engaged as a Research Assistant with successive undergraduate and postgraduate scholarships at the Human–Computer Interaction (HCI) Laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology—Hellas (FORTH) working on Ambient Intelligence Technologies, Machine learning and User Interface Optimization methods. Her interests have been progressively focusing in the direction of Machine learning applied in the domain of Neuroscience. In this respect, she has participated in research work in the Institute of Molecular Biology and Biotechnology (IMBB) of FORTH, in the context of a Computational Neuroscience project regarding the incorporation of biological dendritic features in Artificial Neural Networks (ANNs). Furthermore, she had a six-month internship at the Laboratory of Cognitive Neuroscience (LNCO) of the École Polytechnique Fédérale de Lausanne (EPFL), working on the development of a Virtual Reality (VR)-based experimental paradigm to study visual hallucinations in Parkinson's disease. She is currently a Ph.D student in the Eberhard Karl University of Tübingen in Germany, working in the intersection of Computer Science and Neuroscience.

GEORGE MARGETIS is a computer scientist specialized in interaction design, Ambient Intelligence Technologies and smart environments, Human Centered Artificial Intelligence, XReality as well as networks and telecommunications. He obtained his PhD in ''Information Systems and Human Computer Interaction'', his M.Sc. in ''Computer Networks and Digital Communications'' and ''Information Systems and Human Computer Interaction'', as well as his B.Sc. from the Computer Science Department of the University of Crete.

He is currently a Post-Doctoral Researcher at the Human-Computer Interaction Laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology Hellas (FORTH). His past research interests and developments included network traffic measurement and analysis in high-speed networks, resource control and service differentiation in wired networks. His current research focuses on interaction design and interactive systems development, Ambient Intelligence, Universal Access and Design for All. His recent Research and Development (R&D) work includes tools and interaction techniques for multimodal interaction, design and development of X-Reality systems, as well as personalization, decision making algorithms and recommendation services in the fields of Ambient Intelligence and smart environments, Human Centered AI design, visual analytics and interactive media. He has been actively involved contributing in the specification, coordination of proposal writing and project management of several European, National and industry funded R&D projects in the above areas. Currently, he is the ICS-FORTH Scientific Responsible of the OPTIMAI (G.A. 958264) and 5G MediaHub (G.A. 101016714) Horizon 2020 European Projects, as well as the Consortium Technical Manager of the DARLENE (G.A. 883297), OPTIMAI (G.A. 958264) and COPA Europe (G.A. 957059) Horizon 2020 projects.

He has co-authored more than 60 scientific peer reviewed publications, including book chapters, articles in journals and in proceedings of international conferences and workshops. He is member of the Program Committee and Paper Review Committee in various international journals, conferences and workshops.

STAVROULA NTOA is a computer scientist specialised in usability engineering, User Experience (UX) research and design, and software accessibility. She holds a Ph.D. in ''Information Systems and Human-Computer Interaction'', an M.Sc. in ''Information Systems'' and ''Computer Networks and Digital Communications'', as well as a B.Sc. from the Computer Science Department of the University of Crete.

Currently, she is a post-doctoral researcher at the Human-Computer Interaction (HCI) Laboratory of the Institute of Computer Science of the Foundation for Research and Technology—Hellas (ICS-FORTH), leading the UX research and design activities of the HCI laboratory. She is experienced in the design, development and evaluation of accessibility software for users with disabilities, and accessible web applications. She has expertise in UX design and evaluation in a number of projects in various contexts and application domains, including responsive web, big data, mobile, intelligent, and X-reality applications. Her research interests focus on adaptive and intelligent interfaces, universal access and accessibility of modern interactive technologies, and user experience research in intelligent and Artificial Intelligence environments.

She has been conducting UX scientific and applied research in more than 40 European National and industry funded projects. Currently, she is the quality control responsible and lead UX researcher in five H2020 projects, namely OPTIMAI, 5G MediaHub, 5G Epicentre, Copa Europe, and DARLENE. She has co-authored more than 60 publications. She is co-chair of the International Conference on Artificial Intelligence in Human-Computer Interaction (AI-HCI), and editorial board member of the Universal Access in the Information Society International Journal. She is also a member of the program committee and paper review committee in various international conferences and reviewer in prestigious international journals.

GEORGE PAPAGIANNAKIS (Member, IEEE) received the B.Eng. degree (Hons.) in computer systems engineering from The University of Manchester, Manchester, U.K., the M.Sc. degree (Hons.) in advanced computing from the University of Bristol, Bristol, U.K., and the Ph.D. degree (Hons.) in computer science from the University of Geneva, Switzerland, in 2006.

He is currently an Associate Professor at the Computer Science Department, University of Crete, Greece, and associated as a Faculty Member at the Human–Computer Interaction Laboratory, FORTH-ICS, and a Visiting Associate Professor at the University of Geneva. Prior to this post, he contributed as a Lecturer, a Senior Researcher, and a Research Assistant at the MIRALab, University of Geneva, with Prof. Nadia Magnenat-Thalmann. He has also been employed as a lead computer graphics programmer in the industrial virtual reality simulation sector. He is a computer scientist specialized in computer graphics systems, extended reality algorithms, and geometric algebra computational models. He has more than 80 publications in the field, over 1985 citations and H-index 24 (July 2021). In 2017, he has published a Springer-Nature book on *Mixed Reality and Gamification for Cultural Heritage* which achieved more than 70,000 downloads so far, reaching the top 25. His research and development interests are centered in the field of high-fidelity interactive computer graphics systems for human–computer interaction, featuring embodied presence, psychomotor learning, and gamification with simulated virtual humans in extended reality based on geometric algebra computational models. Research questions are driven from experiential medical VR training to virtual heritage, and from geometric algebra GPU computational simulation to intelligent, symbiotic extended reality systems for virtual humans and other big datasets.

Prof. Papagiannakis is a Board Member of the Computer Graphics Society (CGS) and a member of the ACM, Eurographics, and SIGGRAPH professional societies. In 2011 he has been awarded with a Marie-Curie Intra-European Fellowship for Career Development from the European Commission's Research Executive Agency. In 2016, he served as the Conference Chair of the Computer Graphics International (CGI'16) Annual Conference. In 2020 and 2021, he served as a Programme Co-Chair. In 2016, under the auspices of CGI and CGS he co-founded the Empowering Novel Geometric Algebra for Graphics and Engineering (ENGAGE) Workshop running annually ever since. In 2016, he co-founded ORamaVR as a deep-technology spatial computing medical VR company, building the world's most intelligent, symbiotic VR authoring machine for the rapid acceleration of human learning in medicine. He is an Associate Editor of the *The Visual Computer* journal (Springer), a Research Topic Lead Editor of the *Frontiers in Virtual Reality* journal, and an Evaluator/Reviewer of the European Commission and several National Research Funding agencies worldwide.