# Information Model and Architecture Specification for Context Awareness Interaction Decision Support in Cyber-Physical Human–Machine Systems

João Quintas, Paulo Menezes, and Jorge Dias

***Abstract*—This paper aims to contribute to situation, activity, and goal awareness in cyber-physical human–machine systems (HMS) by presenting a new information model and specifications for a decision-making component that can be integrated in current system architectures. The objective of this work is to improve the efficacy, acceptance, adaptability, and overall performance of HMS and human–system interaction (HSI) applications using a context-based approach. Our hypothesis is that we can enhance current interaction functionalities by integrating context and interaction information models into a decision-making component that behaves as a supervision process for controlling interaction. In HSI, we aim to define a general human model that may lead to principles and algorithms, allowing more natural and effective interaction between humans and artificial agents. The approach was implemented and tested targeting application in the domain of active and assisted living. The challenge of user acceptance is of vital importance for future solutions and is still one of the major reasons for reluctance to adopt cyber-physical systems in this domain.**

***Index Terms*—Active and assisted living, adaptive systems, context awareness, decision systems, human–machine systems, human–system interaction architecture.**

## I. Introduction

THIS paper addresses the problem of autoadaptation of interfaces and interaction strategies in human–machine interaction (HMI) by formulating an approach that looks into the complete combinations of components available to implement a functionality and considers context information at each moment to optimize the sequence of components/algorithms that implement a given functionality.

Manuscript received February 28, 2016; revised June 1, 2016 and September 18, 2016; accepted October 27, 2016. This work was partially carried out in the frame work of the CogniWin project, funded by the EU Ambient Assisted Living Joint Program and the Fundaxção para Ciência e Tecnologia (AAL 2013-6-114), as well as the IntegrAAL project, funded by the EU Ambient Assisted Living Joint Program and the Fundação para Ciência e Tecnologia (AAL 2014-1-187). This paper was recommended by Associate Editor L. Chen.

J. Quintas is with the Instituto Pedro Nunes, 3030-199 Coimbra, Portugal, and also with the University of Coimbra, 3000-213 Coimbra, Portugal (e-mail: jquintas@ipn.pt).

P. Menezes is with the University of Coimbra, 3000-213 Coimbra, Portugal (e-mail: pm@deec.uc.pt).

J. Dias is with the University of Coimbra, 3000-213 Coimbra, Portugal, and also with Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: jorge@deec.uc.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2016.2634923

Given the multidisciplinary nature of HMI, it includes problems of cognition (planning and decision making), navigation, action, human–robot interaction (perception, environment sensing, and interfacing with the end-user), and architecture development and middleware. Fong *et al.* [1] analyzed these issues and extended this to socially interactive robots, pointing out the need to extend on human–robot interaction capabilities in order to address issues imposed by social interaction (see also [2] or [3]).

This motivates the creation of such autoadaptation features in cyber-physical systems, particularly when the interfaces and interaction strategies must adapt to the user expectations and characteristics.

In this sense, there has been a shift from low-level raw observation data and their direct/hardwired usage, data aggregation, and fusion to high-level formal context modeling, activity recognition and behavior analysis, and change detection. It is envisioned that this trend will continue toward a further higher level of abstraction, achieving situation, activity, and goal awareness to facilitate enhanced human–machine systems and human–system interaction.

This is particularly relevant, as implementing real-world systems requires handlers to design particular solutions from generic approaches resulting in static architectures. In these architectures, we can assume that the scalability and redundancy of some functionalities are already considered (e.g., security, fault tolerance, and data storage). However, robust approaches for HMI are still a challenging topic. Taking into account some examples from recent works [4]–[9], we can identify key factors that prevent these systems from being ideal in terms of interaction. These factors, among others, are related to uncertainty associated with noise inputs, variation in environment conditions, and unclear expectations from the user. In spite of the sophistication of state-of-the-art systems, we could not find a holistic approach to address HMI. Typically, each functionality is addressed individually and later integrated overall through well-defined and fixed interface protocols. However, the lack of redundancy and fall-back strategies in terms of interaction functionalities often results in unexpected system behaviors (e.g., faults, errors, or fails), creating barriers of adoption to new technologies or new interaction modalities.

The research question being addressed can be stated as: What approach can we follow to achieve autoadaptation in human–system interaction functionalities in cyber-physical systems?

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



In this hypothesis, assume that any architecture can be described as a network topology of algorithms and we can find redundant sequences of algorithms (i.e., paths in the network) that implement the same functionality. In these conditions, autoadaptation will be facilitated by introducing a decision process that considers the context at a given moment to select the best sequence of algorithms, in which requirements are satisfied by the current context.

To test this hypothesis, we set up an experiment based on the example explored in our previous work [10] and extended it considering a typical use case in HMI that involves people detection, as a first step to trigger the interaction. For the selected example scenario, we foresee an improvement in terms of precision and recall of people detection, while observing a decrease in computational time to achieve the results, as a consequence of the selection of the more adequate strategy (i.e., selection between two available algorithms) for a given context.

In Section II, we present a survey of related works. In Section III, we describe the architecture views and specifications. Section IV refers to our analysis and specifications of our information model for human–system interaction. In Section V, we discuss our approach implementation and initial results. Section VI concludes the paper, summarizing the major findings and stating further work.

## II. Related Work

### A. Architectures and Models for HMI

Alami *et al.* [11] presented an integrated architecture allowing a mobile robot to plan its tasks, taking into account temporal and domain constraints, to perform corresponding actions and to control their execution in real time, while being reactive to possible events. The general architecture is composed of three levels: a decision level, an execution level, and a functional level. The latter is composed of modules that embed the functions achieving sensor data processing and effector control. The decision level is goal and event driven, and it may have several layers. According to the application, their basic structure is a planner/supervisor pair that enables users to integrate deliberation and reaction. Alami *et al.* [12] discussed a decisional framework for human–robot interactive task achievement that aimed to allow the robot to produce behaviors that support its engagement vis-a-vis its human partner and to interpret human behaviors and intentions. The architecture used for controlling the robot followed a similar three-layered architecture from their previous work [11], but highlighted some aspects as situation assessment and context management, goals and plans management, action refinement, execution, and monitoring. This generic architecture model has been adopted in state-of-the-art robotic applications; refer, for example, to the SPENCER project by Triebel *et al.* in [13]. On the other hand, state-of-the-art interaction models similar to that proposed by Sili *et al.* [14] typically refer to some degree of adaptation, but explicit models must be provided to rule out the behavior of the system. Their proposed interaction model is depicted in Fig. 1. In this model, the authors consider that control and decision mechanisms are included within the Dialogue Manager component. In recent works, for example,

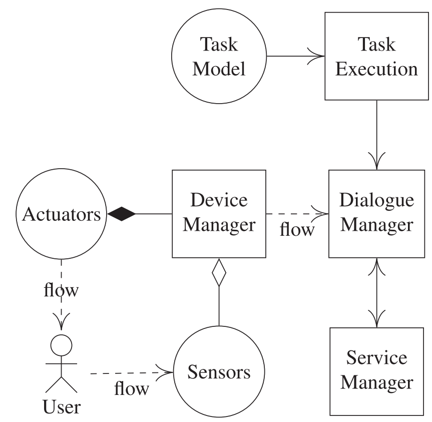



Fig. 1. Interaction model.

the study of Devin *et al.* [15], the authors summarize the essential building blocks to design an architecture for cognitive and interactive robots. The concepts presented may be generalized for human–machine systems overall.

### B. Decision Systems Used in HMI

We can find in the literature recent works that address adaptation processes involved in HMI, but they mainly focus on task planning. Altisen *et al.* [16] formalized a general intermediate layer approach that allowed automatic generation of property-enforcing layers to be used between an application program and a set of resources for which safety properties are defined and should be respected by the global system (the application, plus the intermediate layer, plus the set of resources). Alami *et al.* [17] focused on the organization aspects of the agent decisional abilities and on the management of human interaction as an integral part of the agent control architecture. Their proposed framework allowed the agent to accomplish its tasks and produce behaviors that support its engagement vis-a-vis its human partner and interpret similar behaviors from him. The framework was applied in a companion agent scenario by the same authors [18] within the scope of the Cogniron project. Clodic *et al.* [19] presented the agent control architecture, SHARY, which is dedicated to agent action in the presence of or in interaction with humans. This architecture focused more on task planning but provided support to implement a supervision system adapted to HMI.

### C. Context Definition and Context Models

Context has been studied extensively in language use, usually with "context," meaning the history of prior utterances (see, e.g., [20]), but also including other kinds of context. Holtgraves [21] has found that the status of the speaker relative to the hearer affects whether the literal meaning of an indirect request is activated.

These and other examples have been motivating different scientists to design and implement context-aware systems in a variety of fields of application.

Boytsov *et al.* [22], in reference to Bazire and Brézillon [23], "analyzed 150 definitions of context for different subject areas." The conclusions were that although there is no absolute consensus regarding some aspects of context and its definition, the common understanding is that "context is the set of circumstances that frames an event or an object" and "context acts like a set of constraints that influence the behavior of a system (a user or a computer) embedded in a given task." Moreover, the definitions proposed by Dey [24] are still considered the most popular definitions of context in pervasive computing, which define context as "any information that can be used to characterize situation of an entity."

Schmidt *et al.* [25] proposed to develop a hierarchically organized feature space for context. At the top level, they proposed to distinguish between context related to human factors in the broadest sense and context related to the physical environment. Both general categories are further classified into three categories. The six categories at this level provide a general structure for context. Within each category, relevant features can be identified, again hierarchically, whose values determine context. Additional context is provided by history, that is, by changes in the feature space over time.

Human-factor-related context is structured into three categories: information on the user (knowledge of habits, emotional state, biophysiological conditions, etc.), the users' social environment (colocation of others, social interaction, group dynamics, etc.), and the users' tasks (spontaneous activity, engaged tasks, general goals, etc.). Likewise, context related to physical environment is structured into three categories: location (absolute position, relative position, colocation, etc.), infrastructure (surrounding resources for computation, communication, task performance, etc.), and physical conditions (noise, light, pressure, etc.).

The types of context according to Abowd *et al.* [26] are location, identity, activity, and time. These were considered primary context types for characterizing the situation of a particular entity. The primary pieces of context for one entity can be used as indices to find the secondary context for that same entity as well as the primary context for other related entities. In this categorization, they proposed a simple two-tiered system. The four primary pieces of context are on the first level. All other types of context are on the second level. The secondary pieces of context share a common characteristic: they can be indexed by primary context because they are attributes of the entity with primary context.

Bisgaard *et al.* [27] analyzed previous definitions and models up until the mid-2000s. The survey summarizes articles representing the general body of the literature on context awareness in human–computer interaction. Their conclusions resulted in four major context features: Location, Time, Identity, and Environment; a complimentary group of features include: Social setting, Network, Season, History, Task/Activity, and Device. However, when mapping their application in context-aware systems, they concluded that a smaller context space is used, typically defined by three context features. The resulting "workable" context features would be fused into a group of five: Location, Identity, Time, Environment, and Activity.

Bazire and Brézillon [23] concluded that "context occurs like what is lacking in a given object for a user to construct a correct representation." Some determining factors that define a context model are as follows: the entity/subject concerned by the context, its focus of attention, its activity, its situation, its environment, and, eventually, an observer.

Bettini *et al.* [28] described the state of the art in context modeling and reasoning that supports gathering, evaluation, and dissemination of context information in pervasive computing. Most prominent approaches to context modeling and reasoning are rooted in database modeling techniques and in ontology-based frameworks for knowledge representation. They also presented state-of-the-art techniques to deal with high-level context abstractions and uncertainty of context information. The survey finally introduced hybrid approaches as an attempt to combine different formalisms and techniques to better fulfill the identified requirements.

### *D. Summary*

Based on the information from our survey, we conclude that we commonly find approaches that treat context and interaction models separately resulting in monolithic architectures for components related to interaction functionalities. This aspect could be improved by exploring what approaches can successfully integrate both models and how they could be implemented.

Furthermore, the most accepted definitions for context are those proposed by Schilit *et al.* [29], Pascoe [30], and Dey [24], which can be summarized into a general notion that Context is defined as all information that characterizes a situation. The problem with this definition is that it is difficult to understand what information characterizes a situation. Moreover, the typical examples used to refer to context result as description for a place or an event (e.g., in the context of Kitchen, in the context of Meeting). This makes it hard to generalize the definition to other applications, for which their information domain was not previously formalized, modeled, and represented.

This problem applies to human–machine systems and human–system interaction. We identified this challenge as an obstacle to our aim of achieving enhanced interaction functionality in such systems. A suitable model and formalization of context and interaction concepts are required for decision algorithms that supervise HMIs.

## III. Context-Based Architecture for Interaction Activity Management

Following the structure of the generic architecture proposed by Alami *et al.* [11], we can identify particular components that are directly involved in the interaction process, as previously presented by Quintas *et al.* [10]. Taking into account established approaches, we consider that each system or machine offers a set of functionalities that can be decomposed into a set of algorithms, working under a given context.

### *A. Discovering Connections Between Algorithms*

Let us consider the case of high granularity, which we analyze at the algorithm level. The system is the set of algorithms, and each algorithm has restrictions on the input data and the output

**for** $i, j = 1$ **to** $N$ **do**
  **if** $Ain_i = Aout_j$ **then**
    $d_i, j \leftarrow True$
  **else**
    $d_i, j \leftarrow False$
  **end if**
**end for**

Fig. 2. Algorithm to establish network of the system.

data. A given sequence of algorithms results in the implementation of a functionality.

We consider the availability of algorithms that provide redundancy (i.e., same inputs and outputs), but are optimized to address the same problem in different contexts. On the other hand, we also consider the availability of algorithms that can be sequenced (i.e., first algorithm output information to the second). Following these considerations, we can establish various possible sequences to implement the same functionality, depending on the conditions for a given time (i.e., context).

By chaining some of these algorithms, we can expect to progressively transform the input data into the desired information, for example, personal identity and location. Such a chaining of algorithms can be viewed as a set of nodes in a graph connected with arcs that represent the data that flow from one algorithm output to the subsequent input.

These sequences of algorithms may assume, therefore, a network topology, requiring us to first formulate the network model that describes the possibilities of the relationships between the algorithms.

Our adaptation problem can, thus, be summarized as the search for a method that, from all possible sequence combinations, outputs the optimal sequence of algorithms to implement a given functionality, depending on the conditions prevailing at each instant.

We formalize the system by means of an interconnection of functionalities, thus forming a network of algorithms.

Let us assume that we have $N$ algorithms $A_i$, where $i = 1, \ldots, N$, which are characterized by their inputs $Ain_i$ and outputs $Aout_i$. Two algorithms have a dependence if the input $Ain_i$ of $A_{i+1}$ is the same of the output $Aout_i$ of $A_i$.

We can now establish that if we have a finite set of algorithms and their description in terms of their inputs and outputs, we can establish the network that represents the systems architecture, using the algorithm shown in Fig. 2.

For the algorithm description, consider $A_i$ and $A_j$ as two algorithms; the dependence $d_i, j$ exists if $Ain_i$ is the same as $Aout_j$, where $i$ and $j$ are integer numbers between 1 and $N$, in which $N$ is the number of algorithms available in the system.

The result of this algorithm will give the basic structure for the network that connects all possible combinations of available algorithms in our system, which will be similar to that represented in Fig. 3. In this figure, the square nodes represent sensors, the circles represent the algorithms, and the triangle represents the goal we want to achieve.

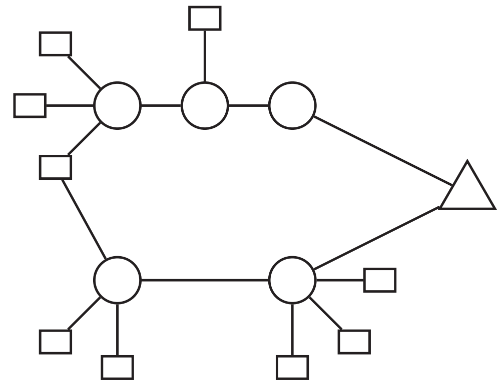


Fig. 3. Network topology for sequenced algorithms.

## B. Introducing Dynamic Changes

With the previous representation established, we can now focus on how to incorporate the context-based mechanism that will allow us to select the optimal sequences of algorithms in the case the system is affected by condition changes, which may result in errors or faults. We want to be capable of switching between algorithms that maximize the chance of achieving a desired goal. This adaptation avoids replanning. The optimization of algorithm selection depending on changing conditions corresponds to a decision process.

Consider that each algorithm that processes direct or transformed sensor data has some requirements for it to work properly. These requirements may be related to the values contained in the input data that must be within some optimum range, or some other specific characteristics. As an example, we can mention a vision-based simultaneous localization and mapping process that only may work if there is adequate lightening for extracting the required features from the camera images.

These requirements form a set of constraints that correspond to context. When reaching the limit conditions set by the requirements, we must decide if we should switch to an alternative that either provides similar functionality or resets context, thus putting the decision process as part of system context-based adaption.

If an algorithm requires conditions $a$, $b$, and $c$ (e.g., context1) to work, but for some reason, one or more of these cannot be verified, the decision process will check if the system can perform an action in such a way that context1 can be presented and the algorithm can be applied. This can happen primarily in two ways: first, the system could find some way to influence the environment in such a way that the conditions/context for the algorithm is satisfied; second, the system could find some internal strategy to overcome such a limitation—for example, it could use an auxiliary algorithm that can be used with the current context or it can generate the context for the functionality that must be used (see Fig. 4).

At this stage, we need an optimization approach that can control the selection of the most promising sequence to achieve a goal. Considering this, we need to establish the costs/weights of each edge, which will be associated with the context perceived

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



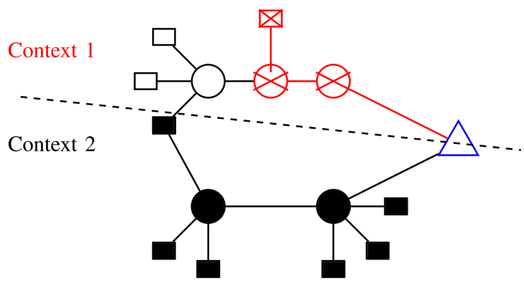



Fig. 4. Selecting the sequence for the given context.

in a given moment. For each context, the set of edges will have a different cost. This information is passed to decision analysis, which finally decides the optimal sequence of algorithms (i.e., path) to achieve a goal (i.e., provide a functionality).

This approach allows us to incorporate redundancy in terms of HMI by guaranteeing multiple possible solutions to achieve a similar result (ideally, the same) adapting to different contexts of operation.

### C. Modeling the Decision-Making Process

Typical approaches used in decision analysis include decision trees [31], influence diagrams [32], multicriteria decision making [33], or Markov chains [34].

For our problem, we need an approach that takes into consideration aspects regarding limitations in *a priori* planning (i.e., we cannot plan every possible course of actions *a priori*) and the limited capability of measuring the state of the world (i.e., limited perception capability). Consequently, this will introduce uncertainty in the decision process, because a decision must be taken based on incomplete information.

In this work, we extend our previous work introduced in [10] and [35] and will adopt partially observable Markov decision processes (POMDPs) to address the above issues. A POMDP is described as the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, T, R\}$ that can be specified as follows.

1) $\mathcal{S}$—State is the way the world currently exists. This set represents all possible information about the agent and its context (e.g., location and environment conditions).
2) $\mathcal{A}$—Actions form the set of possible alternative choices you can choose to make, which include algorithms that can be executed to provide a certain functionality.
3) $\mathcal{O}$—Finite set of observations of the state of the world, which correspond to measurable parameters (e.g., sensor readings). In our model, context is included in the decision model as variables in the set of observations.
4) $\Omega(a, s, o) : \mathcal{O} \times \mathcal{S} \times \mathcal{A}$—This captures the relationship between the state and the observations (and can be action dependent). $\Omega(a, s, o)$ tells the agent the probability that it will perceive observation $o$ when in state $s$, after performing action $a$. To define the observation function, we consider a set of conditional probabilities
$$Pr(o|s', a). \tag{1}$$
5) $T(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$—The transition function, or the likelihood of transition from state $s$ with action $a$ to new state $s'$. To define the transition function, we consider a set of conditional probabilities
$$Pr(s'|s, a). \tag{2}$$
6) $R(s, a) : \mathcal{S} \times \mathcal{A}$—The reward function; this refers to the reward received for transitioning to state $s$ with action $a$. We specify some immediate value for performing each action in each state. The reward or payoff function in POMDPs is defined as
$$r(b, a) = \sum_s b(s) r(s, a). \tag{3}$$

*a) Belief Update:* The agent can then use the observations $o$ it receives to update its current belief $b$. Specifically, if the agent's current belief for state $s$ is $b$, it takes action $a$ and gets an observation $o$, then its new belief vector $b'$ can be determined using
$$b'(s') = \frac{\Pr(o|s', a) \sum_{s \in \mathcal{S}} \Pr(s'|s, a) b(s)}{\sum_{s' \in \mathcal{S}} \Pr(o|s', a) \sum_{s \in \mathcal{S}} \Pr(s'|s, a) b(s)} \tag{4}$$
where $b(s)$ is the value of $b$ for $s$.

*b) Policy:* The solution to a POMDP is called a policy, and it simply specifies the best action to take for each of the states. We will use $\pi$ to denote the agent's policy. The optimal policy can, thus, be defined as
$$\pi^*(b) = \arg\max_{a \in \mathcal{A}} \left[ r(b, a) + \sum_{o \in \mathcal{O}} P(o|b, a) V^*(b') \right] \tag{5}$$
where $\pi^*(b)$ yields the highest expected reward value for each belief state $b$, represented by optimal value function $V^*$, where $b'$ is the next belief state of the agent.

## IV. Human–System Interaction Information Model

The integration of context and interaction information models is a vital aspect in achieving situation, activity, and goal awareness to facilitate enhanced human–machine systems and human–system interaction.

After we have described our approach to address architectural aspects, we now focus on the manner in which we address the logical representation of the previous model.

### A. Functionality–Algorithm–Information Models

In our approach, to achieve system automatic adaptation to dynamic interaction processes, we must link functionalities and algorithm properties to our information model. It is particularly relevant to establish the relationships between context entities and the interaction entities.

First, we consider functionality description. Each functionality is usually associated with at least one algorithm. For each functionality, the description should include properties Parameters and Type referring to generic information about the class. The association with the class Context relates a given functionality to its requirements in terms of context entities

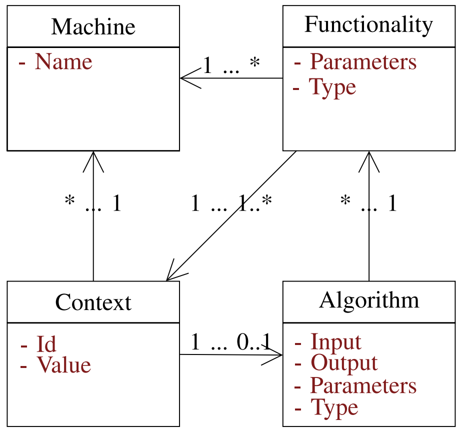



Fig. 5. Functionality description model.

(i.e., the conditions when it can be executed). At this level, instances of Context may refer to concepts given by sources of information other than sensors (e.g., features obtained from a classification process).

Second, we consider algorithm description. The model for any algorithm includes Input, Output, and Parameters. In our description, we used Input and Output to define the input and output datatypes, respectively, and the Parameters to describe the information related to the algorithm type and variables used. As for functionality, the association with the class Context relates the algorithm to the context when it can be applied. At this level, the context instances may refer to concepts given by the outputs of sensors (e.g., range of light intensity where the algorithm is known to work or not).

The logical view of this association is depicted in Fig. 5.

### B. Context Model

Looking back to the definitions discussed in Section II-C, we can identify some key concepts that can be associated with some branches of mathematics (e.g., set theory, vector spaces, and constraint optimization). This is particularly true when the majority of the definitions refer to context as a set of circumstances that act like a set of constraints on the system's behavior (i.e., output). On the other hand, the concept of situation appears associated with context when this is defined as elements of the situation that should impact behavior.

We propose to extend on previous definitions as follows:

> *Context will be defined as the set of information which constrains the performance of an agent while attempting to execute a desired behavior. In spite of the characteristics of that agent, any set of information will be only considered to be context if it anticipates how the agent should behave when that information is present.*

This definition describes context as a set of information that is relevant for decision making, which will result in a specific behavior. After the decision is made, for a behavior to be performed, it is implied that the context is maintained.

In our context model, we took into account the most used context entities by context-aware applications. Therefore, in spite of those identified in the literature—the four main primary context entities (i.e., location, time, identity, and environment) and the six complimentary context entities (i.e., social setting, network, season, history, task/activity, and device)—only five may be considered as "workable" context entities.

We concluded from our survey that the most significant context entities are Location, Identity, Time, Environment, and Activity.

### C. Human–System Interaction Information Model

The previous information models only refer to a part of the knowledge representation required to capture the information domain of human–system interaction. A complete model should also include concepts and entities that are relevant to the model's interaction process. For example, concepts referring to sensors, actuators, robots, or interaction modalities must be taken into account.

Our complete model extends the integration of previous models with these elements, which allow our system to take the entire interaction process into account. A summarized view of our model is depicted in Fig. 6.

## V. Scenario Definition and Experiment Implementation

The experimental implementation sets its baseline in the example explored in our previous work [10] and extends it considering a typical use case in HMI that involves people detection, as a first step to trigger interaction.

The objective of the experiment we address in this work was to understand if our approach would result in a more effective and efficient strategy to detect a human. Given that we are choosing between different algorithms to cover different working conditions, we expected that we would detect a human in more situations, using simple algorithms instead of focusing on complex implementation of data fusion and customized adaptation. In this case, we were interested in finding the number of correct detections and the number of iterations (i.e., computational time) elapsed until a detection occurs.

### A. Experimental Setup

The goal was to detect people under scenarios with varying light conditions and in different backgrounds—thus, in different contexts.

We selected two algorithms—Haar-like features-based detection and histograms of oriented gradients (HOG)-based detection—that are commonly used for people detection. These two approaches provide a similar functionality, but their performance differs according to the conditions of the image (i.e., light intensity).

The experiment used the INRIA dataset for people detection, composed of 288 images with varying number of people per image, in different landscapes and under different light conditions.

The results were obtained from running each algorithm separately and running concurrently. We compared the results from the algorithms selected by our decision process. We measured precision, recall, f-measurement, and computational time for the different runs.

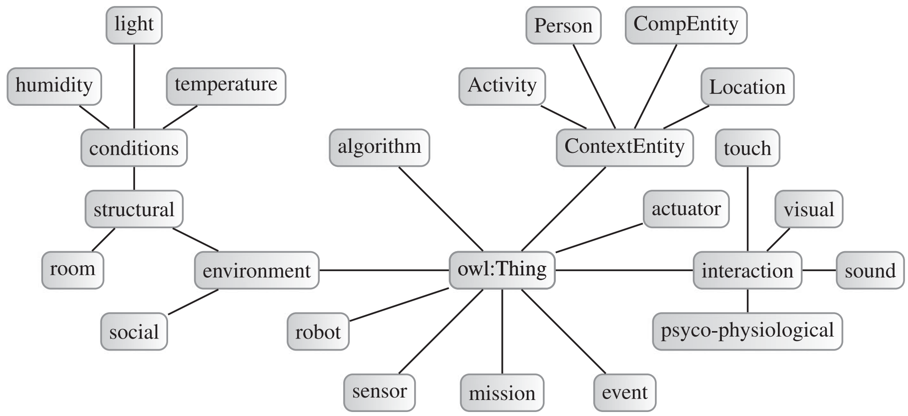



Fig. 6. Taxonomy for the entities in the human–system interaction information model.

TABLE I
STATE TRANSITIONS

| ACTION | START-STATE | END-STATE | P |
|---|---|---|---|
| haar | person-detected | person-detected | 0.90 |
| haar | person-detected | person-not-detected | 0.10 |
| haar | person-not-detected | person-detected | 0.10 |
| haar | person-not-detected | person-not-detected | 0.90 |
| hog | person-detected | person-detected | 0.90 |
| hog | person-detected | person-not-detected | 0.10 |
| hog | person-not-detected | person-detected | 0.10 |
| hog | person-not-detected | person-not-detected | 0.90 |
| check-light | person-detected | person-detected | 1.00 |
| check-light | person-detected | person-not-detected | 0.00 |
| check-light | person-not-detected | person-detected | 0.00 |
| check-light | person-not-detected | person-not-detected | 1.00 |

TABLE II
OBSERVATION PROBABILITIES

| ACTION | STATE | OBSERVATION | P |
|---|---|---|---|
| haar | person-detected | dark-light | 0.05 |
| haar | person-detected | normal-light | 0.50 |
| haar | person-detected | bright-light | 0.45 |
| haar | person-not-detected | dark-light | 0.80 |
| haar | person-not-detected | normal-light | 0.10 |
| haar | person-not-detected | bright-light | 0.10 |
| hog | person-detected | dark-light | 0.45 |
| hog | person-detected | normal-light | 0.50 |
| hog | person-detected | bright-light | 0.05 |
| hog | person-not-detected | dark-light | 0.10 |
| hog | person-not-detected | normal-light | 0.10 |
| hog | person-not-detected | bright-light | 0.80 |
| check-light | person-detected | dark-light | 0.50 |
| check-light | person-detected | normal-light | 0.00 |
| check-light | person-detected | bright-light | 0.50 |
| check-light | person-not-detected | dark-light | 0.50 |
| check-light | person-not-detected | normal-light | 0.00 |
| check-light | person-not-detected | bright-light | 0.50 |

TABLE III
REWARDS

| ACTION | START-STATE | END-STATE | OBS | R |
|---|---|---|---|---|
| haar | person-detected | * | * | 20 |
| haar | person-not-detected | * | * | −100 |
| hog | person-detected | * | * | 20 |
| hog | person-not-detected | * | * | −100 |
| check-light | person-detected | * | * | −5 |
| check-light | person-not-detected | * | * | −5 |

### B. *Defining POMDP Components*

The POMDP relies on defining the set of states, the expected observations from those states, the action transition matrix, and the reward structure.

In our experiment, without restricting generalization, we consider a simple example that we defined with the sets:

$\mathcal{S} =$ person-detected, person-not-detected

$\mathcal{A} =$ haar, hog, check-light

$\mathcal{O} =$ dark-light, normal-light, bright-light

and Tables I–III for transition function $T(s, a, s')$, observation probabilities $\Omega(a, s, o)$, and rewards $R(s, a)$, respectively.

For this experiment, we assumed that our observations were given by measuring the light intensity from a particular image, which we considered to be the explicit result of action check-light. Each observation resulted from the classification of the respective image histogram into the three classes defined in $\mathcal{O}$.

### C. *Results and Discussion*

The results obtained revealed that, initially for the selected dataset, both algorithms work correctly in 80% of the situations, and in the remaining 20%, at least one fails in detecting people (19% one fails, 1% both fail).

The results for HOG features detection showed that it is more precise than Haar-like feature detection overall (49.6% more precise, 27% increased recall, and improvement of f-measurement of 43.5%). However, for the cases of low-light conditions, HOG detection performed poorly, not being capable of detecting in most of the cases (failing in 12% of the dataset). On the other hand, the same applies for Haar-like features detection for cases with high-light conditions (failing in 9% of the dataset). Fig. 7 illustrates an example of the results obtained in the dataset.

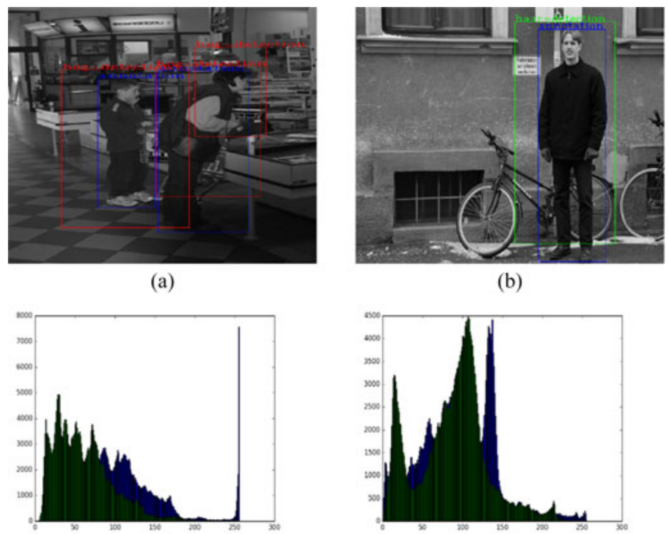



Fig. 7. Example of images with detected persons and image histograms. HOG detection performing better in brighter images and Haar detection performing better in darker images.

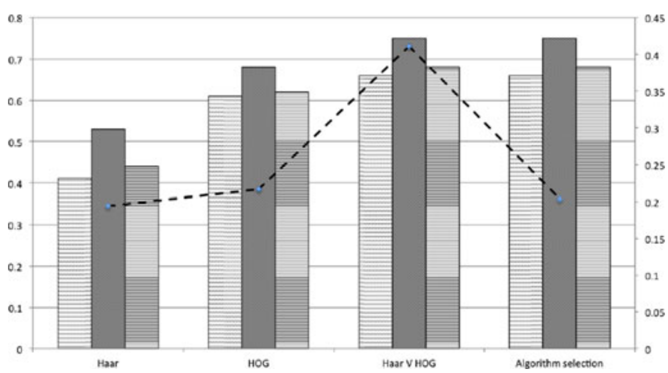



Fig. 8. Experimental results for the different approaches. Bars represent the results for precision, recall, and f-measurement. Dashed line represents computational time taken to complete detection.

For the cases where algorithms made at least one detection, we analyzed computational time, precision, recall, and f-measurement. The results are depicted in Fig. 8.

Attending to the obtained results, we conclude that when performing the experiment using both algorithms running concurrently (i.e., logical OR), we were able to improve the precision in 8.1%. However, this approach incurs added computational time (increase of 89.5%), as one algorithm executed after the other lengthened both computational times. Our approach mitigates this problem (increase of 4.7%), while achieving equivalent improvements in terms of precision and recall (precision = 66%, recall = 75%). The lower computational time results from the decision process, selecting the optimal algorithm to be executed for a given image. The computational time until detecting a person includes the cases when the selected action is check-light, but this can be neglected compared to the computational time required for executing each algorithm.

In spite of the results described above, we must identify limitations for the approach that we followed in this experiment. Given that our results support only the applicability of our approach to an experimental setup for the most simple case, we assume that for some simple examples like in our example scenario, we could have achieved similar results by using threshold for light conditions. However, for more complex systems, this approach would be difficult to implement because it would require expert analysis and it would be impractical for systems with a large set of conditions. On the other hand, the proposed approach inherits the limitations often identified for POMDP. These limitations are associated with a greater computational complexity for models that consider more than two states or that require continuous spaces, which result in unfeasible implementations. However, provided these limitations are solved, for example, by using distributed programming, this approach benefits from being scalable and flexible to change. The model can be adapted to incorporate new variables and trained to generate updated policies (i.e., course of actions). Extrapolating our case study for more complex setups, the improvement in performance and autonomy will prove to be significant.

## VI. Conclusion and Future Work

This paper has presented a decision system that is being developed to improve the HMI process using a context-based adaptation approach. We identified HMI as a vital aspect in the adoption of active and assisted living solutions. Moreover, we identified key HMI features that are commonly referred to by end-users as a must, which are still imposing relevant scientific and technical challenges. To address these challenges, we implemented a system that addresses a key interaction functionality, detecting a user to start the interaction. The innovative aspect of our approach is that of being capable of adapting to changes in the context of operation by selecting the most adequate action (e.g., algorithm) that allows the system to continue to execute with similar performance level, instead of complete failure. The results showed that our approach achieved encouraging results and will pave the course for the next steps in terms of system description representation and automatic discovery of features (i.e., algorithms) available that lead to dynamic composition of interaction skills.

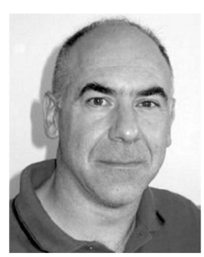
## References

[1] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," in *Robot. Auton. Syst.*, vol. 42, nos. 3/4, Mar. 2003, pp. 143–166.
[2] K. Severinson-Eklundh, A. Green, and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," *Robot. Auton. Syst.*, vol. 42, nos. 3/4, pp. 223–234, Mar. 2003.
[3] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends Human-Comput. Interaction*, vol. 1, no. 3, pp. 203–275, Feb. 2007.
[4] J. Quintas, L. Almeida, M. Brito, G. Quintela, P. Menezes, and J. Dias, "Context-based understanding of interaction intentions," in *Proc. 21st IEEE Int. Symp. Robot Human Interactive Commun.*, Sep. 2012, pp. 515–520.
[5] C. Tsiourti *et al.*, "A virtual assistive companion for older adults: Design implications for a real-world application," in *Proc. SAI Intell. Syst. Conf.*, Sep. 2016, pp. 556–566.

[6] D. Portugal, L. Santos, P. Alvito, J. Dias, G. Samaras, and E. Christodoulou, "SocialRobot: An interactive mobile robot for elderly home care," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2015, pp. 811–816.
[7] G. S. Martins, L. Santos, and J. Dias, "Towards a context-aware adaptable services framework with applications in elderly care," in *Proc. Workshop Improving Quality Life Elderly Using Robotic Assistive Technol., Int. Conf. Social Robot.*, Oct. 2015, poster presentation.
[8] G. S. Martins, L. Santos, and J. Dias, "The GrowMeUp project and the applicability of action recognition techniques," *in Proc. 3rd Workshop Recog. Action Scene Understanding*, Sep. 2015, poster presentation.
[9] S. Kleanthous *et al.*, "Analysis of elderly users' preferences and expectations on service robot's personality, appearance and interaction," in *Human Aspects of IT for the Aged Population. Healthy and Active Aging*. New York, NY, USA: Springer, Jul. 2016, pp. 35–44.
[10] J. Quintas, P. Menezes, and J. Dias, "Auto-adaptive interactive systems for active and assisted living applications," in *Technological Innovation for Cyber-Physical Systems*. New York, NY, USA: Springer, Apr. 2016, pp. 161–168.
[11] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *Int. J. Robot. Res.*, vol. 17, no. 4, pp. 315–337, Apr. 1998.
[12] R. Alami, A. Clodic, R. Chatila, and S. Lemaignan, "Reasoning about humans and its use in a cognitive control architecture for a collaborative robot," in *Proc. Cogn. Archit. Human-Robot Interaction Workshop*, Mar. 2014, pp. 6–9.
[13] R. Triebel *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Proc. 10th Conf. Field Service Robot.*, Jun. 2015, p. 21.
[14] M. Sili, M. Gira, M. Müllner-Rieder, and C. Mayer, "Interaction model-based user interfaces: Two approaches, one goal—Comparison of two user interface generation approaches applying interaction models," in *Information and Communication Technologies for Ageing Well and e-Health*. New York, NY, USA: Springer, May 2015, pp. 185–197.
[15] S. Devin, G. Milliez, M. Fiore, A. Clodic, and R. Alami, "Some essential skills and their combination in an architecture for a cognitive and interactive robot," in *Proc. 2nd Workshop Cogn. Archit. Social Human-Robot Interaction*, Mar. 2016, pp. 1–5.
[16] K. Altisen, A. Clodic, F. Maraninchi, and E. Rutten, "Using controller-synthesis techniques to build property-enforcing layers," in *Proc. 12th Eur. Conf. Program. Program.*, Apr. 2003, pp. 174–188.
[17] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Proc. Joint Conf. Smart Objects Ambient Intell.: Innovative Context-Aware Services: Usages Technol.*, Oct. 2005, pp. 81–85.
[18] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Toward human-aware robot task planning," in *Proc. AAAI Spring Symp., Boldly Where Human-Robot Team Gone Before*, Jul. 2006, pp. 39–46.
[19] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila, "SHARY: A supervision system adapted to human-robot interaction," in *Proc. 11th Int. Conf. Exp.Robot.*, Jul. 2009, vol. 54, pp. 229–238.
[20] E. C. Ferstl, "Context effects in syntactic ambiguity resolution: The location of presuppositional phrase attachment," Ph.D. dissertation, Univ. Colorado, Aug. 1994.
[21] T. Holtgraves, "Communication in context: Effects of speaker status on the comprehension of indirect requests," *J. Exp. Psychol.*, vol. 20, no. 5, pp. 1205–1218, Sep. 1994.
[22] A. Boytsov, A. Zaslavsky, E. Eryilmaz, and S. Albayrak, "Situation awareness meets ontologies: A context spaces case study," in *Proc. 9th Int. Interdisciplinary Conf. Model. Using Context*, Nov. 2015, pp. 3–17.
[23] M. Bazire and P. Brézillon, "Understanding context before using it," in *Proc. 5th Int. Interdisciplinary Conf. Model. Using Context*, Jul. 2005, pp. 29–40.
[24] A. K. Dey, "Understanding and using context," *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Feb. 2001.
[25] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Comput. Graph.*, vol. 23, no. 6, pp. 893–091, Dec. 1998.
[26] G. D. Abowd, A. K. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proc. 1st Int. Symp. Handheld Ubiquitous Comput.*, Sep. 1999, pp. 304–307.
[27] J. Bisgaard, M. Heise, and C. Steffensen, "How is context and context-awareness defined and applied? A survey of context-awareness," Dept. Comput. Sci., Aalborg Univ., Aalborg, Denmark, Jan. 2004, pp. 31–40.
[28] C. Bettini *et al.*, "A survey of context modelling and reasoning techniques," *Pervasive Mobile Comput.*, vol. 6, no. 2, pp. 161–180, Apr 2010.
[29] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proc. Workshop Mobile Comput. Syst. Appl.*, Dec. 1994, pp. 85–90.
[30] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Proc. 2nd IEEE Int. Symp. Wearable Comput.*, Oct. 1998, pp. 92–99.
[31] M. Terek, "Decision trees and Influence diagrams in decision analysis," *J. Appl. Math., Statist. Informat.*, vol. 1, no. 1, pp. 121–135, Apr. 2005.
[32] M. Diehl and Y. Y. Haimes, "Influence diagrams with multiple objectives and tradeoff analysis," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 3, pp. 293–304, May 2004.
[33] H. Fazlollahtabar and M. Saidi-Mehrabad, "Optimizing multi-objective decision making having qualitative evaluation," *J. Ind. Manage. Optim.*, vol. 11, no. 3, pp. 747–762, Jul. 2015.
[34] G. Su, T. Chen, Y. Feng, D. S. Rosenblum, and P. S. Thiagarajan, "An iterative decision-making scheme for Markov decision processes and its application to self-adaptive systems," in *Proc. 9th Int. Conf. Fundam. Approaches Softw. Eng.*, Apr. 2016, pp. 269–286.
[35] J. Quintas, P. Menezes, and J. Dias, "Context-based decision system for human-machine interaction applications," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2016, pp. 3906–3911.

**João Quintas** received the M.Sc. degree in electrotechnical and computers engineering in 2009 from the University of Coimbra, Coimbra, Portugal, where he currently working toward the Ph.D. degree.

He is also a Project Manager and Researcher at the Instituto Pedro Nunes, Coimbra. His main research interests include human–robot interaction, context awareness, behavior analysis, and active and assisted living applications.



**Paulo Menezes** received the Ph.D. degree in electrical and computer engineering with specialization in informatics from the University of Coimbra, Coimbra, Portugal.

He is an Assistant Professor at the University of Coimbra, where he is also a Senior Researcher at the Institute of Systems and Robotics. He develops research activities in areas such as computer vision, human–machine interaction/human–robot interaction, immersive and interactive environments, and human behavior and emotional analysis. He has been involved in several national and European R&D projects in these areas.

**Jorge Dias** received the "Agregação" (Habilitation) degree and the Ph.D. degree in electrical engineering from the University of Coimbra, Coimbra, Portugal, with specialization in control and instrumentation.

He is an Associate Professor at the University of Coimbra, where he conducts his research activities at the Institute of Systems and Robotics. His current research interests include robot vision, computer vision, human–robot interfaces, and social robots, with activities and contributions in these fields. He has been the Principal Researcher in several European research projects.