Context-Based Decision System for Human-Machine Interaction Applications

João Quintas
ISR - University of Coimbra
Coimbra, Portugal
and
Institute Pedro Nunes
Coimbra, Portugal
Email: jquintas@ipn.pt

Paulo Menezes
ISR - University of Coimbra
Coimbra, Portugal
Email: pm@deec.uc.pt

Jorge Dias
ISR - University of Coimbra
Coimbra, Portugal
and
Khalifa University
Abu Dhabi, UAE
Email: jorge@deec.uc.pt

Abstract—In this paper we present a decision process to auto-adapt and improve human-machine interaction, simplifying the integration of algorithms and functionalities. The decision process is part of an innovative approach that integrates contextual information to orchestrate behaviours of an interactive system (i.e. perception and actuation features involved during interaction). Classical approaches focus on designing and implementing algorithms that take into account several environment features (e.g. light, pose, etc.) to adapt its performance obtaining accurate results. An advantage of these approaches is to concentrate complexity in one algorithm leading to simple system architectures. In the other hand, a disadvantage of such approaches is their limitation to adapt to conditions under different scenarios, which typically requires manual adjustments to compensate changes of environment features. Our hypothesis is that, we can improve the overall performance of human-machine interaction process if a decision process is introduced, which is responsible for selecting the most adequate actions/algorithms, with maximum performance, that achieve a certain goal under a given context. The results from exploratory simulations validate the proposed approach to be more effective in attaining specific goals in the interaction process, resorting to algorithms with low complexity.

Index Terms—Human-Machine Interaction, Context-awareness, Decision processes, Human detection, Auto-adaptive interfaces.

I. INTRODUCTION

Within the current trends in agentics we find the willingness to bring agents out of laboratory conditions. This means making them adaptable to the variability that human and/or natural environments introduce in terms of signals that can be acquired about them and that must be converted into valuable information but also on the physical spaces, structures and objects where agents must evolve and/or interact with. Obviously the actuation must be adapted to each situation.

For example, a wheeled agent for indoors use should have wheels replaced for outdoor operation and is unable to evolve on sandy or muddy terrains. But, even if we consider a task of driving an autonomous vehicle from one point, i.e. a system that was designed to evolve on urban roads, it cannot rely on an initially defined trajectory without continuous adaptation. Other vehicles will be present, a road may be interrupted, an obstacle may force a deviation, and even a solar eclipse may occur changing the lighting conditions with respect to those considered during the preparation of the initial plan. Coming back to indoors human environments, any kind of agent cannot rely on its initial plan as the conditions may also change during the task execution. Humans tend to move things from place to place, leave objects on the floor, open window blinds, switch lights on and off, in sum introduce large amounts of changes in the environment that are not predictable. In both cases, these unpredictable situations are true obstacles to the deployment of traditional agents as their functionalities are dependent on sensing modalities and algorithms that cannot adapt to every possible situation. Apart from the changes that people constantly introduce in their environments, other sources of problems may be found related with the architecture of human spaces. In houses, offices, and other buildings, we have rooms, and corridors that use artificial light, where other have windows or glass panels instead of walls for receiving sunlight and provide spectacular views over the outside spaces. Where the spaces with artificial light can be seen as more stable in terms of illumination, and therefore the consequent acquisition of images via camera sensors, the latter may create more complicated and uncontrollable problems for the sensing tasks of agents. Here the lighting conditions vary constantly and may go from dark at night to excessive light in some situations that may saturate the used cameras. As another possible problem, a glass panel as an outer wall may be totally undetectable for some laser-based sensors. To circumvent the problem of this last example we can opt for some ultrasound sensors, but these are far less precise than laser range finders (LRFs), thus a possible rule could be: “use LRF to detect walls except in rooms A B and C (those with glass panels) where sonar sensors and appropriate algorithms must be used”. Yet another example, in Human-Machine Interaction (HMI) applications, multi-modal perception is often required to deliver a set of advanced features commonly identified as needs by the end-users of such applications. In scenarios where human-machine collaboration is foreseen requiring face recognition or gesture interaction, it is imperative first that the user may be detected. This means, visual human detection is a “must have”
functional requirement, with particular relevance in the fields of application of Social agentics and Active Assisted Living. Therefore, we foresee that improving the detection of the user in typically unstructured environments, where the HMI process may be affected by uncertain conditions, have a significant impact on the overall performance of systems.

A. Related works

Multimodal socially-apt agent cognitive systems would allow for both verbal and nonverbal (i.e. emotion- and body language-based) human-like dialogue enactment. This way, the desires and needs of end-users would be recognized and consequently addressed much more efficiently. Service agents, whether capable of social interaction or not, must address many common design problems. The analysis of this issue presented by Fong et. al. [1] is still relevant today; according to these authors, these problems include cognition (planning, decision making), navigation, action, HMI (perception, environment sensing, interfacing with the end-user) and architecture development and middleware.

Human-Machine Interaction technologies and correspond- ing cognitive capabilities of artificial agents have seen many developments in the last few decades - see for example [2] for an extensive survey. Solutions for multisensory active perception and attention allocation have greatly evolved (see, for example, [3]), as has multimodal human emotion and dialogue analysis and human-like emotion and dialogue synthesis (see, for example, [4]) and also human behaviour analysis (see, for example [5]). On the other hand, strategies for dealing with proxemics (e.g. acceptability, etc.) and also safety issues regarding the agent autonomy in a human environment are also still a matter of concern. [2] These facts restrict most of the current socially interactive agents to highly controlled environments and specialised applications.

State of the art interaction models [6] typically refer to some degree of adaptation but explicit models must be provided to rule out the behaviour of the agent. Furthermore, we can find in the literature, recent works that address adaptation processes involved in HMI, but they mainly focus on task planning. In [7] the authors formalized a general intermediate layer approach, which 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). In [8] the authors 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 in [9] within the scope of the Cogniron project. In [10] presented the agent control architecture SHARY, dedicated to agent action in presence or in interaction with humans.

This architecture focused more in task planning but provided support to implement a supervision system adapted to HMI.

B. Summary

Attending to the state-of-the-art, we conclude that proposed approaches disregard aspects related with the dynamics of task execution. This means, after task planning is concluded the system may have different paths to choose from, but the decision process associated with the control of task execution in limited to a set of prior rules, which may or may not correspond to the optimal action to take for a specific situation. Moreover, typical approaches tend to design and implement algorithms that take into account several environment features (e.g. light, pose, etc.) to adapt its performance obtaining accurate results. An advantage of these approaches is to concentrate complexity in one algorithm leading to simple system architectures. In the other hand, a disadvantage of such approaches is their limitation to adapt to conditions under different scenarios, which typically requires manual adjustments to compensate changes of environment features and do not allow to incorporate additional capabilities (i.e. add new algorithm to operate in new conditions).

C. Contribution and paper organization

Our hypothesis is that, we can improve the overall performance of HMI system if a decision process is introduced, which is responsible for selecting the most adequate algorithms, with maximize performance to achieve a certain goal under a given context.

Our work presents a decision process to auto-adapt and improve HMI, simplifying the integration of algorithms and functionalities, thus extending previous work in [5], [11]–[13]. This approach makes it possible to select which algorithm or functionality that works or provides better performance under a certain set of circumstances (i.e. context). With our approach, for a given set of planned goals, we can decide the course of actions that maximize achieving the goal with maximum performance. In case certain conditions change unexpectedly our approach can adjust automatically in run time which action to take whilst contributing to achieve the setup goal.

In the following sections, in II we present the foundations of our work, in section III we define a case study experiment of our approach to achieve a context-based human detection feature for HMI. In sections IV and V we present our experimental results, we discuss about them and conclude with some closing remarks.

II. METHODOLOGY

A. Adaptation through context awareness

The above presented situations suggest the inclusion of a decision process that takes into account the present context for the selection of the sensors and algorithms to apply at each instant. The trivial solution seems to define this decision layer based on a set of rules that define for each context which sensor and/or algorithm to use. This requires a predefinition of those rules together with the relevant contexts where they should be applied.
1) Predicted versus observed context: The definition of the context that influences the execution of tasks by an autonomous robot may include, for the sake of improved autonomy, observable quantities that could extend the definition of context beyond the typical time, location, task, etc. This would permit to define modes of operation given observable and thus dynamic contexts, besides the static ones. Including a true context awareness in the decision process enables the adaptation of the robot to changing work conditions that may appear at unpredictable times and places.

B. Context-based Decision System

In case task execution 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 re-planning. The optimization of algorithm selection depending on changing conditions corresponds to a decision process.

Each algorithm will only work correctly if its requirements are met. These requirements form a set of constraints that correspond to context. When reaching to limit conditions set by the requirements, we must decide if we should switch to an alternative that either provides similar functionality or resets the current state of the world either. This consideration introduces uncertainty in the decision process, because a choice must be made based on incomplete information. Nevertheless, we would like to describe the fact that the agent does not know in which state it is in but, instead, believes that it can be in any number of states with certain probability.

Partially Observable Markov Decision Processes (POMDPs) address the above issues. Instead of directly observing the current state, the state gives us an observation that provides a hint about what state it is in. The components involved in the formalisation of a POMDP are described as the tuple \( \{S, A, O, T, B, \Omega, \} \) that can be specified as:

- \( S \) - represents the state of the world. The variables included in this set represent all possible information about the agent and its context (e.g. location, environment conditions).
- \( A \) - within the set of actions we include the algorithms that can be executed to provide a certain functionality.
- \( O \) - finite set of observations corresponding to measurable parameters (e.g. sensor readings). In our model, context is included in the decision model as variables the set of observations.
- \( \Omega(s, a, o) \) - the relationship between the state and the observations (and can be action dependent). \( \Omega(s, a, o) \) tells the agent the probability that it will perceive observation \( o \) when in state \( s \), after performing action \( a \).
- \( T(s, a, s') : S \times A \times S \) - the likelihood of transition from state \( s \) with action \( a \) to new state \( s' \). The transitions specify how each of the actions change the state. To define the transition function

\[
\tau(b, a, b') = \sum_{s'} \Omega(a, s', o) \sum_s T(s, a, s') b(s) \tag{1}
\]

- \( R(s, a, s', o) : S \times A \times S \times O \) - reward function, reward received for transitioning from state \( s \) to \( s' \) with action \( a \) and observation \( o \). If we want to automate the decision making process, then we must be able to have some measure of an action’s cost or state’s value so that we can compare different alternative action policies over the long term. We specify some immediate value for performing each action in each state. The reward or payoff function in POMDPs is defined as

\[
r(b, a) = \sum_s b(s) r(s, a) \tag{2}
\]

1) Belief update: The agent can then use the observations it receives to update its current belief \( b \). Specifically, if the agent’s current belief is \( b \) and it takes action \( a \) then its new belief vector \( b' \) can be determined using

\[
b'(s') = \alpha \Omega(a, s', o) \sum_s T(s, a, s') b(s), \tag{3}
\]

where \( b(s) \) is the value of \( b \) for \( s \) and \( \alpha \) is a normalizing constant that makes the belief state sum to 1.
2) Policy: The solution to an 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 i’s optimal policy can thus be defined as

$$\pi^*(b) = \arg \max_{a \in A} R(b,a) + \sum_{o \in O} P(o|b,a)V^*(\tau(b,a,o))$$

(4)

where $\pi^*(b)$ correspond to the computation of a value function over the belief space.

III. IMPLEMENTATION AND EXPERIMENTAL CASE STUDY

In order to test our approach we considered a typical use case in Human-Machine interaction that involves people detection, as a first step to triggers the interaction.

The objective of this experiment was to understand if our approach would result in a more effective and efficient strategy to detect a human. Given we choose between different algorithms to cover different working conditions, we expected 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 number of iterations (i.e. computational time) elapsed until a detection occur.

A. Experimental setup

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

We selected two algorithms that are commonly used for people detection, but they work in complementary and similar conditions with different performances.

The experiment used the INRIA dataset for people detection, composed by 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. After, we compared with the results from the algorithms selected by our decision process. We measured precision, recall, f-measurement and computational time for these different runs.

B. Visual human detection based on appearance

1) Haar-like features based detection: Based of rectangular features similar to Haar basis functions. Simple features that can be computed very rapidly using an integral image. Although these methods are very efficient, the Haar-like features are not capable of handling the complexity in real-world images and videos.

2) Histograms of Oriented Gradients (HOG) based detection: The advantage of HOG features is that each image cell is statistically represented by a histogram of the gradient orientations and magnitudes, thus it is more invariant to illumination, shadows, etc.

Light conditions is known to have a strong influence in visual-based people detection, hence it is a known limitation. We setup an experiment where the agent has the capability of executing the two detection algorithms Haar-like features detection and HOG features detection, but it must be capable of selecting the one that ensures better performance based on light conditions.

C. 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 as: 

![Context-based decision process workflow](image)
Our 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 systems, the improvement in performance and autonomy will prove to be significant.

V. CONCLUSION AND FUTURE WORK

In this paper we presented a decision process to auto-adapt and improve human-machine interaction, simplifying the integration of algorithms and functionalities. Our approach makes it possible to select which algorithm or functionality that works or provides better performance under a certain set of circumstances (i.e. context). We tested this approach with an experimental case of people detection. We selected two algorithms that are commonly used for people detection, but they work in complimentary and similar conditions with different performances. We compared the precision, recall and F1 measurement of the different performances of the algorithms running in the INRIA dataset for people detection. The obtained results confirm our hypothesis for a simple approach which can be defined in such a way that a positive reward is given to an action that leads to the state of a person detected, but penalizes otherwise. Penalties were set as to encourage the action of checking context (i.e. light conditions). 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 systems, the improvement in performance and autonomy will prove to be significant.

IV. RESULTS AND DISCUSSION

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

Running the algorithms separately we obtained fair results, with HOG features detection being more precise that Haar-like features detection overall (49.6% more precise, 27% increased recall and improvement of F-measurement of 43.5%). In the cases of low light conditions, HOG detection performed poorly, not being capable of detecting most of the cases (corresponding to fail in 12% of the dataset). On the other hand, we obtained similar results for Haar-like features detection for the cases with low light conditions (corresponding to fail in 9% of the dataset). In figure 2 we illustrate an example of the results obtained in the dataset.

For the cases where algorithms made at least one detection, we analysed computational time, precision, recall and f-measurement. The results are depicted in figures 3a to 3d.

Attending to 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, running both algorithms in the same image concurrently incurs additional computational time (increase of 89.5%) that can be avoided by our approach (increase of 4.7%), when compared to individual results. Our approach was able to achieve equivalent improvements in terms of precision and recall (precision = 66%, recall = 75%), but with less computational operations, as it selects the algorithm (i.e. action) that provides better performance according the light intensity in the image (i.e. context) at each time.

On the other hand, for the selected example scenario, we could have achieved similar results using threshold for light conditions. However, for more complex systems that approach would be difficult to implement because it would require expert analysis and it would be impractical for systems which a large set of conditions.

Our 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.

\[ R(s, a, s', o) = \begin{cases} 10 & \text{if } s = s, \text{ and both } O_{a1} = 1, O_{a2} = 0, \text{ and } O(s, o) = 10 \\ -10 & \text{if } s = s, \text{ and both } O_{a1} = 1, O_{a2} = 0, \text{ and } O(s, o) = 10 \\ 1 & \text{otherwise} \end{cases} \]

\[ R(s, a, s', o) = \begin{cases} 10 & \text{if } s = s, \text{ and both } O_{a1} = 1, O_{a2} = 0, \text{ and } O(s, o) = 10 \\ -10 & \text{if } s = s, \text{ and both } O_{a1} = 1, O_{a2} = 0, \text{ and } O(s, o) = 10 \\ 1 & \text{otherwise} \end{cases} \]
yet important use case, which encourage us to extend this approach to more complex setups.

REFERENCES


