Hardware and Software Structure for a Social Robot Capable of Situation Analysis

N Kimr1, N Bodunkov1[0000-0002-5485-3362] and J Sinyavskaya1[0000-0002-1142-4648]

1Moscow Aviation Institute, Moscow, Russia

E-mail: nkim2011@list.ru

Abstract. This paper discusses structuring of hardware and software for an autonomous social robot. It shows that the real-world social robot operations face the challenge of environmental variability and uncertainty of the objective parameters. Thus, a social robot must be capable of situation analysis for better autonomy. We propose a modular distributed structure of the control system. Separate modules monitor the status and control the subsystems of the robot. General coordination of subsystems is provided by the Supervisor module. For the robot to function autonomously, the Supervisor must be capable of situation analysis and its key functions: objective retrieval and analysis, situation description, configuring and strategizing the solution. The robot’s sensory inputs help acquire the objective and its parameters to describe the situation. Description relies on the database of a priori knowledge of the environment and its objects. Analysis is linked to a reduction in the uncertainty of the objective parameters and situation description. For a case study, the paper demonstrates a maze-solving strategy as affected by the situation.

1. Introduction
Robots are ever more prevalent in human life. Beside industrial applications, robots are an increasingly popular choice for household, medical, transport, entertainment, and other uses. Social robots capable of autonomy and interaction with humans are another actively evolving application. They are becoming part of socially significant activity (healthcare and education), entertainment, and advertisement. There are two groups of social robots: toys and assistants.

Paro, Fujitsu Bear, MiRo, etc. belong to the former group [1-3]. They are semi-stationary bots equipped with sensors and designed to facilitate patient rehabilitation and to monitor the condition of the elderly and children. Their interaction with humans is limited to responding to touch and dialog.

Unlike this group, robots from the latter group (Pepper, Kiki [4-6]) feature a mobile platform (they are not stationary), arms (to perform their functions and/or add emotions), and a speech interface for voice interaction. Beside the basic smart functions of the former group, these robots can move and interact with their users and space more actively, whether it is emotional or physical interaction. This greatly expands their potential applications. Such robots can serve as tour guides that enrich their speech with gestures; they can fetch items or administer medicines to patients. A robot must be able to do all of this autonomously in a real-world setting, which may involve presence of other humans.

It is therefore of practical significance to carry out research towards better autonomy of social robots in their objectives as set in the real world. Autonomy is defined herein as the ability to complete an objective without a human operator, i.e., in a situation where the user is not trained to operate the
robot. This creates a need to develop approaches that would enable robots to function in uncertain situations (including those where the objective parameters are uncertain) and to adapt to change in the environment (e.g., to reroute if there is an obstacle) [7]. At the same time, a robot must be generally capable of monitoring its own status and be safe for human users.

We propose an approach based on situation analysis as the key tool of adaptation, uncertainty reduction, and combined completion of various objectives. Situation analysis herein refers to the making and analyzing the descriptions of: objectives, objects in the environment and their relations, and the robot’s internal status. Through such analysis, the robot must be able to complete its objective (defined herein as the task or problem the unit is intended to solve). At the same time, its hardware and software must be capable of situation retrieval and analysis, i.e., the robot must feature the required sensors, controls, and computing.

The paper discusses MASHA, a social robot jointly developed by Moscow Aviation Institute and Arzamas Instrumentation Plant, as a case of social robot’s hardware and software.

2. **Social robot and how it can complete its objectives**

A social autonomous robot (SAR) is designed for a variety of objectives, whether it is household or medical assistance. MASHA is conceptualized to follow a particular scenario for each objective. For instance, one scenario describes the delivery of items to the user (a patient). These could be household items, medicines, or devices, e.g., medical instruments. This objective breaks down into the following steps:

- Receive and recognize the user’s voice command,
- Find the required object,
- Grip the object,
- Move to where the user could reach the SAR,
- Hand the object over to the user.

Each step involves several complex local procedures.

This is the most versatile scenario. Some of its procedures could be used with other objectives. Beside these objectives, the robot always has to ensure the safety of its own and its environment. Safety-related objectives include collision avoidance, prevention of human hazard, monitoring the internal status, etc. Control parameters and priorities of these objectives cannot always be preconfigured.

Thus, the software kernel of the robot must be able to evaluate the current situation and the predicted situations to eliminate uncertainties and distribute priorities as well as to determine coordinated controls in order to effectively complete its scenarios.

This is where situation analysis becomes fundamental. Situation analysis [8, 9] seeks to describe the situation in order to gain insight in the relationships of the factors essential in the context of the robot’s objectives. Analysis classifies the current and predicted situations to find the most effective control actions in the context of objectives.

The proposed situation analysis technology uses:

1. Premade databases (DB) and knowledge bases (KB) that contain data on the possible objects (phenomena and processes) of interest, their attributes, the rules of their casual relations, and typical objectives;
2. Situation modeling and assessment functions that use readings from the robot’s sensors and other data sources as well as DB and KB entries;
3. Situation analysis models and criteria that help strategize upon how to complete the objectives using the situation models.

Beside the objective description that lists the effectiveness criteria and limitations, the key model of situation analysis include:

- status models for the robot and its subsystems, which determine the effectiveness of objective completion, in particular the specifications such as the handling capacity of the robot’s arms, the mobile platform speed, the field of view of the visual sensors, etc.;
• environment models describe the state of the environment, i.e., they can be room maps;
• the object-of-interest models depend on the objective. Objects of interest can be items the robot should fetch when asked so by the user.

Situation analysis can be complicated if the environment is variable; the objectives are many and diverse; coordinated control of all subsystems in a robot is challenging. All of this sets the bar for the onboard hardware and software very high.

3. Hardware and software structure
For modeling, analysis, and elimination of uncertainty, hardware must include the following onboard subsystems:
• a computer vision system as the main source of data on objects, their parameters and relations;
• a speech interface to acquire an objective and more data to finetune the models, e.g., by virtue of robot-user dialog;
• arms, a wheeled mobile platform, a ‘face’ screen on the head and an information screen on the chest, which provide means for the user and the environment to interact with the robot.

Figure 1 shows the modular hardware structure that includes all these subsystems.

![Figure 1. Hardware structure of Masha, a social robot.](image)

Onboard computer needs to be a high-performance unit in order to complete the objectives, coordinate the subsystems, analyze the situation, and continuous monitor the robot’s internal status in real time.

Modularity helps mitigate the computational intensity, expand the functionality, improve the usability and maintainability. As a principle, modularity implies that each subsystem is an independent software and hardware module or a set thereof. Each individual module controls some of the robot’s components or systems: its arms, mobile platform, etc.

Modules can independently run complex control algorithms. For instance, the mobile platform module can independently route, navigate, and control the movement. The intelligent module based on a Jetson NANO onboard computer processes images, recognizes objects, etc. Beside the objective function, modules monitor their internal statues and respond to special situations in real time.
Apparently, the factors of the current situation may alter the objective or limit the available solutions. Thus, the structure needs a module to analyze the situation and coordinate the entire set. This is done by a RaspberryPi. Modules are linked by network interfaces. The robot mainly uses RS-485 and Eth.

Software is modular as well. Various critical functions run in separate modules. Figure 2 shows the software structure.

There are five core software modules:
1. Dialog module recognizes and synthesizes speech in order to receive voice commands, retrieve objectives, finetune the objective or situation parameters, and interact with the user.
2. Vision is provided by a group of image processing and analysis modules. These are a major source of data for situation modeling. However, since they take a lot of computing resources, they cannot be allowed to function simultaneously. Objective analysis is what prioritizes the modules and their parameters.
3. System control modules comprise software that controls the arms, the mobile platform, the front panel (“the face”), etc. These implement the key control laws for their respective subsystems and process special situations. They can also provide additional data for the situation model.
4. GUI (on PCs or tablets) and onboard interface (the chest screen) are auxiliary modules for low-level interaction with the robot.
5. Supervisor is the core software module that coordinates all the subsystems in the context of the current situation (environmental conditions and the robot’s status) and the scenario (Turn on the TV, Fetch, Answer Questions, etc.). It is the Supervisor that analyzes the situation. The Supervisor:
   1. Analyzes the situation (is it standard or non-standard situation?) and decides on further actions (“Proceed with the objective” or “Stop” if the situation is unsafe);
   2. Receives queries and commands from the user interface;
   3. Monitors own status (the operability of subsystems);
   4. Monitors the completion of tasks under the running scenario;
   5. Detects special situations and outlines appropriate response;

Figure 2. Software structure of the social robot.
6. Predicts situations and schedules other subsystems’ actions.

4. Implementation of situation analysis components

To complete an objective as commanded by the user or appropriate for the situation, a robot can follow multiple scenarios at a time. For instance, it can be simultaneously busy with monitoring its status, ensuring safety, and competing the objective, etc. The Supervisor is tasked to start or stop individual scenarios, prioritize them, and determine which data is needed to run the scenarios (i.e., to eliminate uncertainty). In fact, the Supervisor acts as a situation analyst.

The objective is what triggers situation analysis routines. The Supervisor receives the objective from the speech channel, an interface command, or the robot’s data storage if it is a recurrent objective.

It tracks the main sources of commands and objectives, describes them, stores them in memory, and analyzes them. Objective analysis can be broadly understood as uncertainty assessment and elimination. Paper [10] uses entropy as a versatile criterion of uncertainty. There are three ways to eliminate uncertainty:

1. Use known data from the DB.
2. Use situation descriptions. Since the process is designed to finetune a specific parameter, the final description will only cover the factors, objects, and relations essential for this specific objective.
3. Take an action, i.e., influence the situation (move, converse, or apply physical action to an item).

Situation models can have various renditions. In general, the situation model has an hierarchical structure. Paper [11] proposes using semantic networks to connect models that vary in nature and rendition. In particular, part of an objective description can be written as:

\[ G((SO(1)):Type(Book(n)): /*the objective is to find a book, shape n*/

The object’s attributes are described as:

\[ AtrM(H(15)^L(50)^W(20))... /*Shape*/ ^AtrObs(B(Not)^C(G)^T(Not)^F(Not))^... /*Observability*/ \]

What complicates modeling is the need to describe uncertain or changing factors whereby many parameters cannot be measured, the relations (co-influence) of different factors are uncertain, and in particular the causal relationships are difficult to detect, etc. Modeling process uses black box technologies or methods based on known physical laws that generally enable a more accurate description of the investigated processes “in greater” compared to a black box.

Then production rules stored in the DB are used to describe the predicted situations on the basis of the specified current situation models. Here is a fragment of the description of a predicted situation coded (150):

\[ Pr_S(N(150),CS(i(8),j(12)):P(0,85)^T(40)^...,... \]

which specifies the robot strategy code (i) and the object-of-interest code (j).

Given the interaction of all factors and strategies, the predicted situation model estimates the probability of completing the objective (P), the process time (T), and other desired parameters.

If there are several alternative control scenarios, the robot can switch between optimistic (aggressive) and pessimistic (cautious) behavior depending on the objective and on the situation.

For instance, if the robot needs to circumvent an obstacle that completely obstructs the view behind it, the objective can be considered nature play. The robot’s controlled strategies include right-hand circumvention, left-hand circumvention, rerouting, or cancelling the objective. Nature strategies include no passage, difficult passage along a specific route, free passage to the left or to the right. Different factor combinations can have different consequences, see Figure 3.
Figure 3. Obstacle circumvention flowchart.

Before entering the maze, the robot (R) sees two optional routes. Visible part of the left-hand route (1) ends with a wall that goes deep into the maze; the visible part of the right-hand route (2) is seen as a dead end.

We propose using the Hurwitz criterion, which can adjust for the worst and for the best behavior of nature, to find a tradeoff control strategy for the robot. The solution is determined by a combination of minimum and maximum gain \( \alpha_{ij} \). The best strategy is the one that maximizes the criterion \( S \) where

\[
S = \max_i [(1 - \alpha) \max_j \alpha_{ij} + \alpha \min_j \alpha_{ij}],
\]

where \( \alpha \) is the degree of optimism. \( \alpha_{ij} \) is the gain where the UAV follows the ith strategy and the opponent (nature) follows the jth strategy.

At \( \alpha = 1 \), the Hurwitz criterion converts into Wald’s maximin criterion (extreme pessimism), which is the right-hand route in Fig. 3; at \( \alpha = 0 \), it matches the maximax criterion (extreme optimism), which is the left-hand route.

Apparently, in this particular case maximin returned a more efficient route than maximax.

5. Conclusions
The paper discusses the development of autonomous social robots. Such robots need to be designed to complete their objectives in an uncertain situation and to adapt to a changing environment. Situation analysis-based control is proposed herein to improve autonomy.

The paper further proposes a hardware and software structure for social robots. The structure is modular, and each subsystem consists of independent hardware and software modules. Each individual module controls some of the robot’s components or systems: its arms, mobile platform, etc. Modules are linked by network interfaces RS485 and Ethernet. This effectively implements a distributed control system.

Status monitoring and module coordination are provided by the Supervisor module. The Supervisor is designed to function as a situation analyst. Situation analysis refers to receiving and analyzing the objective. Objective analysis produces a description of the current situation. Situation description outlines the objective, describes the environment and the robot’s status. As a result, objective parameters are finetuned, and a strategy is picked.

This paper presents maze-solving strategy as a case study. It shows how the system picks between an optimistic and a pessimistic strategy depending on the robot’s status.
6. References

[1] Christopher J 2011 Calo, Nicholas Hunt-Bull, Lundy Lewis, and Ted Metzler: Ethical Implications of Using the Paro Robot with a Focus on Dementia Patient Care In: Proceedings of the Thirty-One AAAI Conference on Artificial Intelligence: Papers from the 2011 AAAI Workshop (WS-11-12) pp 20-24

[2] Katz L 2010 Fujitsu goes from laptops to teddy bear bots https://www.cnet.com/news/fujitsu-goes-from-laptops-to-teddy-bear-bots/

[3] Dowdy C 2017 MiRo the £600 robo-dog is part butler, part carer with an added 'sixth-sense' https://www.wired.co.uk/article/miro-robo-dog-elderly

[4] Nao (robot) Homepage https://en.wikipedia.org/wiki/Nao_(robot)

[5] Robot KIKI Homepage https://robotkiki.ru/

[6] Broadbent E, Jayawardena C, Kerse N, Stafford R, MacDonald B 2011 Human-Robot Interaction Research to Improve Quality of Life in Elder Care - An Approach and Issues. In: Proceedings of the Thirty-One AAAI Conference on Artificial Intelligence: Papers from the 2011 AAAI Workshop (WS-11-12) pp 13-19

[7] Amiri S, Shirazi M S, Zhang S 2020 Learning and Reasoning for Robot Sequential Decision Making under Uncertainty In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence pp 2726-2733

[8] Endsley M, Bolte B and Jones D 2003 Designing for Situation Awareness: An Approach to User-Centered Design New York: Taylor & Francis pp 312

[9] Kim N and Bodunkov N 2015 Adaptive surveillance algorithms based on the situation analysis Computer Vision in Control Systems 2 eds. M Favorskaya, L Jain (New-York:Springer) 7 pp 169-200

[10] Bodunkov N E, Kim N V, Mikhailov N A 2020 The ground objects monitoring by uav using a search entropy evaluation Advances in Intelligent Systems and Computing 1226 AISC pp 443–452

[11] Bodunkov N E, Pindiiurina M O 2019 Use of semantic description of reference scene for visual navigation solutions Journal of Physics: Conference Series 1353(1) 012107

Acknowledgement

This research has been conducted with financial support from the Russian Foundation for Basic Research (RFBR) as part of Scientific Project No 19-08-00613 A.