# Developing a software framework for social robots: Some Issues and Experiences

**Minsu Jang · Ho-sub Yoon · Jaehong Kim**
**Human-Machine Interaction Research Group,**
**Intelligent Robotics Research Labs**
**Electronics and Telecommunications Research Institute,**
**South Korea**
**{minsu, yoonhs, jhkim504}@etri.re.kr**

**Abstract** We succinctly introduce some issues we experienced while designing and experimenting with a software framework for social robots. Our framework, inspired by Aldebaran’s naoqi framework, provides a chatbot-oriented interaction system interfaced with multimodal event handling based on rich perception and behavior control features. Building simple interaction scenarios was easy in our framework, but enriching them with more complex and nuanced situations was very hard due to some technical issues like lack of perceptual continuity, false positives in perception and difficulty of implementing multiparty interaction. These issues can downgrade the overall quality of social robot experiences and make engaging people in long-term repetitive interactions very difficult.

**Keywords** Social Robots · Chat-bots · Software Framework

## 1 Introduction

With a recent rapid technological advancement in the field of speech recognition, voice synthesis and language understanding, verbal interaction has finally become a viable and effective communication skill for social robots. As most social robot services are provided along with a series of face-to-face dialog-based interactions, chatbot engines are being regarded as a pivotal element of the social robot operating systems and frameworks. For example, Naoqi, a software framework for Nao and Pepper, employed Chatscript [1], and extended it to accept multimodal sensory events and respond with multimodal robot behaviors, which has finally become QiChat [2]. Chatbot scripts provide a simple framework to develop interactive applications based on a type of event-reaction rules, so it is very effective in building dialog-based service scenarios for social robots.

## 2 SHRI Framework

Inspired by Naoqi’s QiChat, we designed a software framework for social robots, casually called SHRI, pronounced as shee-ree), representing **S**ocial **H**uman-**R**obot **I**nteraction framework. The basic scheme of chatbot-oriented social robot framework is as shown in Fig. 1.

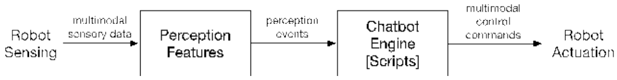

Fig. 1. The basic scheme of chatbot-oriented framework

Perceptual events are passed onto chatbot engines so that multimodal context changes can trigger robot’s response in a verbal as well as non-verbal mode. Traditional chatbots respond passively only to textual input events, but chatbot engines in social robots are extended to be wired into multimodal event channels, and chatbot script languages for social robots include syntactic elements to specify multimodal events and actions.

Based on SHRI, we could build interaction scenarios according to a typical process of rule-based chatbot script programming. The process was intuitive, effective and efficient when building simple interactions, and the result worked well in controlled situations.

But the basic scheme in Fig. 1 showed several limitations when we tried to write scripts that can deal with more complicated situations.

## 3 Some Issues and Experiences

We describe, in this section, the issues we encountered in our experiences with SHRI and our interim solutions to alleviate some of them.

### 3.1 False Positive Perceptions

An interaction with passive devices like smart speakers begins with a user saying hello to it. On the contrary, social robots can be programmed to proactively begin an interaction whenever it detects a person, which may be more natural and user friendly in a social sense.

But in our experiences, false person detections occurred frequently that made our robots greet onto hollow spaces or misplaced objects. This kind of seemingly mechanistic behavior always hindered users from engaging with robots. In some cases, even though a person is correctly detected in a frame, that person gets lost in a later frame and then detected again afterwards. This discrete nature of robot perception easily breaks chatbot-based reactive scripts.

### 3.2 Perceptual Discontinuity

Perception functionalities typically work per image frame accompanied by occasional errors. As such, perception in robots is discrete and noisy. A face recognized as person A at time *t-1* may be recognized as person B at time *t*. If a script sentence is written to say hello to a newly identified user, it may trigger greetings twice which is out of context. To reduce nonsensical responses from robots, we added a functional layer after perception in SHRI that aggregates various perception results for some time duration and deliberately decide when to trigger perceptual events. For example, recognition results are aggregated for 30 frames and the final result is decided by a majority voting mechanism. But it does not alleviate the problem completely, because it is uncertain in many cases which perception result is trustworthy than others. A more elaborative and fundamental mechanism is needed to provide perceptual continuity systematically.

### 3.3 Difficulty of Specifying Multiparty Interactions

In our experiences, implementing multiparty interactions with chatbot script language, an extended rendition of RiveScript [3] in our framework, was always a big challenge. To support multiple users, it was necessary to index each user to manage different interaction states and react accordingly. This is almost unachievable with simplified chatbot script languages with minimal support for programming idioms. If a chatbot script well supports declaring complicated data structures and controlling complex processes, it is no longer a chatbot script. The simplicity of chatbot script makes it possible for domain experts such as consultants and guides to build their own interactive social robot services. If a chatbot script is a full-powered complicated programming language, we have no reason to use it instead of typical programming language.

Eventually, we had not even a roundabout tactics to build multiparty interactions with our script language. We suppose that there are a lot of framework design and scripting issues with regard to supporting multiparty interactions.

## 4 Conclusions

Recent social robots increasingly employ chatbots for the purpose of supporting dialogs and providing an easy way to program multimodal interactions. Building and experimenting with a chatbot-oriented software framework for social robots, we experienced several problematic issues due to incomplete perception features, rule-based reactive mechanisms, and chatbot script languages being too simplistic to support diverse and complicated social robot interactions. We conjecture that these issues should well be addressed to make current social robots succeed in the real world in the long run.

**Acknowledgement** This work was supported by the Technology Innovation Program (10077553, Development of Social Robot Intelligence for Social Human-Robot Interaction of Service Robots) funded By the Ministry of Trade, industry & Energy (MOTIE, Korea)

## References

1. Wilcox, B. (2011). Chatscript. [online] Available: http://chatscript.sourceforge.net.
2. Aldebaran (2014). QiChat. [online] Available: http://doc.aldebaran.com/.
3. Petherbridge, N. (2017). RiveScript, [online] Available: http://www.rivescript.com/.