



# FrameTalk: Human and Picture Frame Interaction Through the IoT Technology

Wen-Shu Lai<sup>2</sup>, Yi-Bing Lin<sup>1</sup>, Chung-Yun Hsiao<sup>1(✉)</sup>, Li-Kuan Chen<sup>1</sup>,  
Chao-Fan Wu<sup>2</sup>, and Shu-Min Lin<sup>2</sup>

<sup>1</sup> Department of Computer Science, National Chiao Tung University,  
Hsinchu 30010, Taiwan, R.O.C.

liny@cs.nctu.edu.tw, phoebe.cyhsiao@gmail.com,  
clikuan.cs05g@g2.nctu.edu.tw

<sup>2</sup> Institute of Applied Arts, National Chiao Tung University,  
Hsinchu 30010, Taiwan, R.O.C.

wndylai@gmail.com, dannylinsam@gmail.com,  
294889468@qq.com

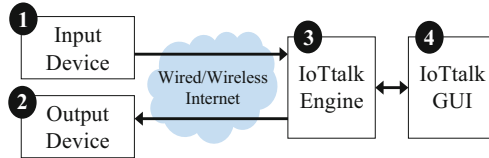
**Abstract.** The concept of frame plays a significant role in art design. However, traditional picture frames do not allow interaction with human. To provide another dimension of artwork demonstration of picture frames, we utilize the Internet of Things (IoT) technology to create human and picture frame interaction. Specifically, an application-layer IoT device management platform called IoTtalk is used to quickly establish connections and meaningful interactions between IoT devices. The frame and human interaction is achieved by implementing an output device of IoTtalk called FrameTalk, which provides animation in a digital frame to be controlled by the input devices (typically a smartphone). The FrameTalk device can be displayed in specific frame hardware or the screen of any computing device. We describe the functional block diagram of the FrameTalk device and its default applications. Then we show two FrameTalk application examples. The first example is CalligraphyTalk that allows the audience to interact with the frame to write the poems. The second example is PortraitGuess that allows the audience to guess historical figures through the frame.

**Keywords:** IoTtalk · Interactive visual design · Cyber physical interaction

## 1 Introduction

Picture frame has been used to demonstrate artworks since ancient times. The importance of the frame was emphasized by a quote of American novelist Chuck Palahniuk: “It’s funny how the beauty of art has so much more to do with the frame than the artwork itself.” Famous Canadian director David Cronenberg used “frame” to encourage the photographers: “That’s rule number one for a photographer, isn’t it? Fill your frame?” Although the concept of frame plays a significant role in art design, traditional picture frames do not allow interaction with human. In recent years, some commercial digital frames can be controlled to display the static image through mobile apps [1] or hand gestures [2]. Some can display dynamic graphics [3, 4]. To provide

another dimension of artwork demonstration of picture frames, this paper utilizes the Internet of Things (IoT) technology to create human and picture frame interaction. Specifically, an application-layer IoT device management platform called IoTalk [5–7] is used to quickly establish connections and meaningful interactions between IoT devices without concerning the lower-layer IoT protocols. Figure 1 illustrates the simplified IoTalk network architecture.



**Fig. 1.** IoTalk network architecture.

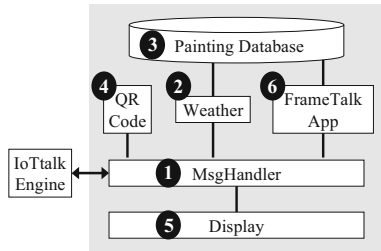
In this architecture, the IoT devices (Fig. 1(1) and (2)) are connected to the IoTalk engine (Fig. 1(3)) in the Internet through wireline or wireless technologies. An IoT device is called an input device if it is a group of sensors (such as color sensor and temperature sensor) or controllers (such as switches and buttons). The input device generates and sends data to the IoTalk engine for processing (see path (1) → (3) in Fig. 1). Similarly, an IoT device is called an output device if it is a group of actuators (e.g., robot arm). The output device receives data from the IoTalk engine (see path (3) → (2) in Fig. 1) to drive its actuators. In IoTalk, every IoT device is characterized by its functionalities or “device features”. A device feature (DF) is a specific input or output “capability” of the IoT device. For input devices such as a smartphone, the input device features (IDFs) are sensors (e.g., the microphone, the Gyroscope, the GPS, and the camera) or controllers (e.g., the keypad). For output devices such as an electric fan, the output device features (ODFs) are actuators including motor rotation, motor speed and so on. A network application defining the interaction between the IoT devices is automatically generated and executed at the IoTalk engine. When the values of the IDFs are updated, an IoT device informs the network application to take some actions, and the network application sends the result to the ODF of the same or another IoT device to affect the actuators of that output device. With this view, the IoT devices interact with each other through their device features. Details of IoTalk operations can be found in [5–7].

The frame and human interaction in IoTalk is achieved by implementing an output device called FrameTalk, which provides animation in a digital frame to be controlled by the input devices. The FrameTalk device is cyber (an animation program) that can be displayed in specific frame hardware or the screen of any computing device. This paper is organized as follows. Section 2 describes the functional block diagram of the FrameTalk device and its default applications. Sections 3 and 4 show two FrameTalk applications and examples for interaction. Section 5 concludes our work with future research directions.

## 2 The FrameTalk Device

A FrameTalk device is implemented as a Java Processing program [8] with the functional block diagram described as follows. The message handler (MsgHandler; Fig. 2(1)) receives instructions from the IoTalk engine through the ODFs of the FrameTalk device. There are two default ODFs. The *Mode* ODF determines which frame application to be displayed. A default application is Weather (Fig. 2(2)) to be discussed later. Another default ODF *Name-O* receives the folder name that corresponds to a painting file folder stored in the database (Fig. 2(3)).

A FrameTalk device is equipped with a QR code scanning mechanism (Fig. 2(4)). When the device detects that a person may want to interact with the frame, a QR code pops up at the right corner at the bottom of the frame display area (Fig. 2(5)). The person uses a smartphone to scan the QR code to enter the webpage of the FrameTalk application. In this way, the smartphone can interact with the frame through its browser without installing any mobile app. We will elaborate more on the mobile app in Sect. 4.



**Fig. 2.** Functional block diagram of a FrameTalk device.



**Fig. 3.** The frame automatically reflects the weather at this moment: (a) sunny with scattered clouds, (b) rainy, (c) cloudy.

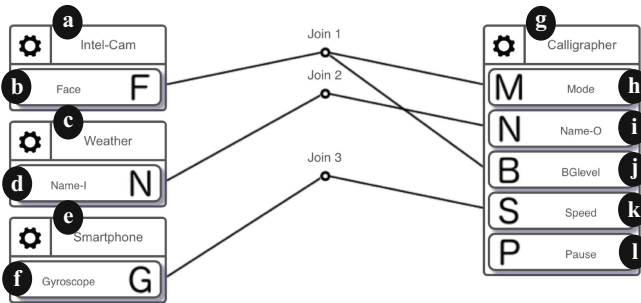
The default display mode is weather-painting, where the frame executes the Weather module to automatically display a painting corresponding to the current

weather condition [9]. For example, the frame displays “Impression, Sunrise” of Monet when it is sunrise with scattered clouds outdoors (Fig. 3(a)). The frame shows “The Umbrellas” by Renoir when it is raining (Fig. 3(b)), and shows “Autumn Landscape with a Flock of Turkeys” by Millet when it is cloudy (Fig. 3(c)). These paintings are stored in the database in Fig. 2(3). The weather information is provided by any input device that can detect or predict the local weather condition. The weather conditions can be 0 (clear sky), 1 (few clouds), 2 (scattered clouds), 3 (broken clouds), 4 (shower rain), 5 (rain), 6 (thunderstorm), 7 (snow), and 8 (mist).

Besides the weather application, we can develop new applications as software modules (Fig. 2(6)) to be integrated in the FrameTalk device. Switching among the applications is controlled by the *Mode* ODF. We will describe the development of two FrameTalk applications in the subsequent sections.

### 3 CalligraphyTalk: A Cyber Calligrapher that Talks

CalligraphyTalk is a FrameTalk application including the following input devices: Weather, Intel-Cam and one or more smartphones. The output device is a FrameTalk device called Calligrapher. IoTalk implements interaction of these IoT devices by configuring the connections in the GUI window (Fig. 1(4)). In this GUI, an input device is represented by an icon placed at the left-hand side of the window (e.g., Fig. 4 (a), (c) and (e)), which consists of smaller icons that represent IDFs (e.g., Fig. 4(b), (d) and (f)). Similarly, an output device is represented by an icon placed at the right-hand side of the window (e.g., Fig. 4(g)), which includes ODF icons (e.g., Fig. 4(h)–(l)). For the purpose of readability, if both an IDF and an ODF have the same name, the GUI will append the IDF name with “-I” and the ODF name with “-O”. For example, the configuration in Fig. 4 includes the *Name-I* IDF and the *Name-O* ODF. Figure 5 illustrates the functional block diagrams of the input and the output devices, and the connections correspond to the configuration in Fig. 4.



**Fig. 4.** The IoTalk GUI for a CalligraphyTalk application.

Besides the weather-painting mode described in Sect. 2, CalligraphyTalk includes the interactive-calligraphy mode. Switching between the two modes is determined by

Intel-Cam (Fig. 4(a)). The hardware of Intel-Cam is an Intel RealSense Camera (model F200 or SR300), which can detect human facial expressions. The detection functions are provided by Intel RealSense SDK [10] as shown in the functional block diagram (Fig. 5(1)). When  $N$  persons stand in front of the frame, their faces are detected, and the *Face* IDF of Intel-Cam sends out the value  $N$  (the interactive-calligraphy mode). If no face is detected, it outputs value 0 (the weather-painting mode). In the weather-painting mode, the Weather input device (Fig. 4(c)) uses IP Geolocation API and OpenWeatherMap API (Fig. 5(2)) [9] to obtain the real-time weather condition (clear sky, raining and so on) of the frame’s location, and output the corresponding folder name of the painting files through the *Name-I* IDF (Fig. 4(d)). The painting files are stored in the calligraphy database (Fig. 5(4)).

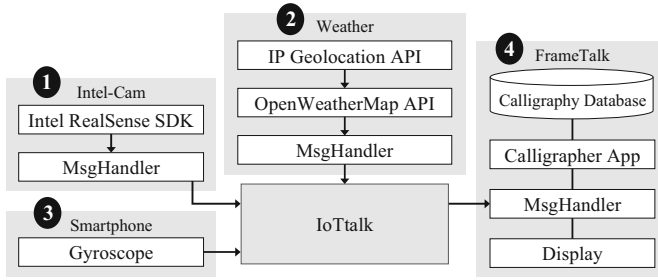


Fig. 5. The functional block diagrams of the CalligraphyTalk application.

The interactive-calligraphy mode is implemented as a FrameTalk application module (Fig. 2(6)), which allows the audiences to interact with the frame to write poems with different speeds. Besides the default ODFs (*Mode* and *Name-O*), Calligrapher implements three ODFs including *BGlevel* (Fig. 4(j)), *Speed* (Fig. 4(k)) and *Pause* (Fig. 4(l)). Through Join 1, the *Mode* ODF is connected to the *Face* IDF of Intel-Cam to determine the display mode of Calligrapher. Through Join 2 the *Name-O* ODF is connected to the *Name-I* IDF of Weather to decide the “weather condition”. The usage of the folder name in the weather-painting mode is described in Sect. 2. In the interactive-calligraphy mode, the folder name is used to determine the background animation pattern. For example, the calligraphy background pattern for a sunny day is shown in Fig. 6. The background pattern for a rainy day is shown in Fig. 7.

The *BGlevel* ODF gives the degree to emphasize the effect of the background pattern. There are three levels (from 0 to 2) as illustrated in Figs. 6 and 7. In the example in Fig. 4, the *BGlevel* ODF is connected to the *Face* IDF of Intel-Cam through Join 1. Therefore, if more people stand in front of the frame, the background effect will be more significant.

The *Speed* ODF (Fig. 4(k)) determines the writing speed of the poem. In our example, the ODF is connected to the *Gyroscope* IDF of a smartphone through Join 3. With the QR code scanning mechanism described in Sect. 2, the smartphone can

connect the Gyroscope sensor to the FrameTalk applications without installing any mobile app. Therefore, any person with arbitrary smartphone can enjoy the interaction with CalligraphyTalk.

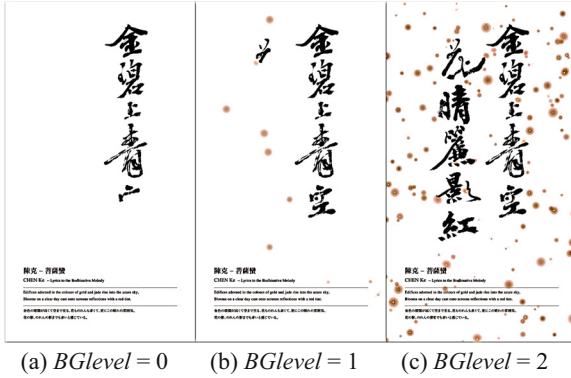


Fig. 6. The  $BGlevel$  for a sunny day.

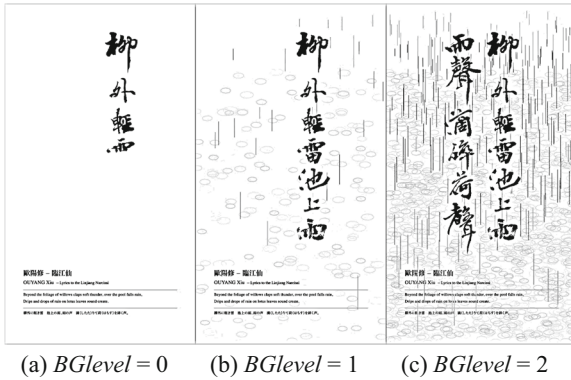


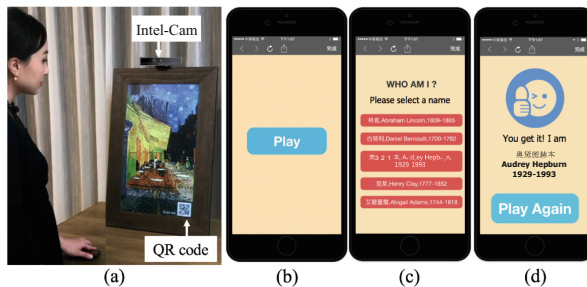
Fig. 7. The  $BGlevel$  for a rainy day.

#### 4 PotraitGuess: Guess Historical Figures

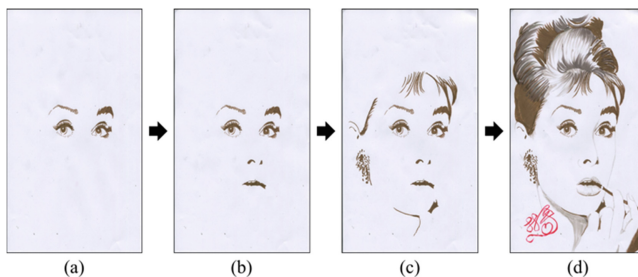
This section describes PotraitGuess, a FrameTalk application that uses a smartphone (the input device) to interact with a digital painting frame (the output device) for portrait guess [9]. Like CalligraphyTalk, PotraitGuess includes two interaction modes with the painting frame: weather-painting and portrait-guess. The portrait-guess mode allows an audience to play a portrait guess game with the picture frame through a smartphone without installing any mobile app. This game has been used in the history classes to guess the historical figures, which significantly attract attention of the students. Figure 8(a) illustrates the frame hardware. We use Intel-Cam to detect if

someone stands in front of the frame. If so, a QR code pops up to be scanned by a smartphone to enter the portrait guess game webpage called PG-web (Fig. 8(b)). In our design, only one smartphone can interact with the frame at a time. Therefore, after a smartphone has scanned the QR code, other people cannot access the QR code anymore to avoid interference of game playing.

When the person presses the “Play” button, the frame shows the first feature of a portrait (Fig. 9(a)). Then the smartphone displays the message “Who am I?” and lists five buttons labelled five names of historical figures to be chosen (Fig. 8(c)). If the person presses a button with a wrong name, the smartphone shows the message “Incorrect answer. Try again!” and the frame displays the next portrait feature (Fig. 9(b)). When the person presses the button labelled with the correct name, the smartphone shows the message “You get it!” and the “Play Again” button (Fig. 8(d)). The frame displays the remaining portrait features one by one (Fig. 9(b) → (c) → (d)), and stops at the final (complete) portrait (Fig. 9(d)). The person presses “Play Again” button to start a new game.



**Fig. 8.** (a) Entering the portrait-guess mode. (b)–(d) The smartphone portrait guess game webpage.



**Fig. 9.** The painting feature displaying.

Figure 10 illustrates the connections of the input and the output devices, where the corresponding functional block diagrams of the IoT devices are illustrated in Fig. 11. The Weather input device (Fig. 10(a)) and the Intel-Cam input device (Fig. 10(c)) are the same as those in Fig. 4. The portrait-guess mode is implemented as a FrameTalk

application module (Fig. 11(2)). This module includes a painting database storing several portrait folders, where each folder has one or more picture files (e.g., (a)–(d) in Fig. 9). In Fig. 10(i), two default ODFs (Fig. 10(j) and (k)) are connected to Weather and Intel-Cam through Joins 1 and 2 just like CalligraphyTalk. PortraitGuess has two extra ODFs besides the default ODFs. The *Forward* ODF (Fig. 10(l)) triggers PortraitGuess to display the next picture in the portrait folder. The *End* ODF (Fig. 10(m)) triggers PortraitGuess to display all remaining pictures in the folder sequentially.

In the PotraitGuess application, the PotraitGuess webpage PG-web (Fig. 8(b)–(d)) is browsed by a smartphone. PG-web in the smartphone (Fig. 11(1)) implements three IDFs including *Name-I* (Fig. 10(f)), *Wrong* (Fig. 10(g)) and *Correct* (Fig. 10(h)). When the audience presses the “Play” button in Fig. 8(b), PG-web selects a portrait name, and sends the name through the *Name-I* IDF to the *Name-O* ODF of PortraitGuess via Join 3 in Fig. 10. When *Name-O* receives the portrait name, the frame shows the first feature of the corresponding portrait (Fig. 9(a)). Then PG-web displays five name buttons. If the audience presses a button with a wrong name, the *Wrong* IDF sends a signal to the *Forward* ODF via Join 4, and the frame displays the next portrait feature. When the audience presses the correct name button, the *Correct* IDF sends a signal to the *End* ODF via Join 5. Then the frame displays the remaining portrait features one by one, and finally stops at the complete portrait.

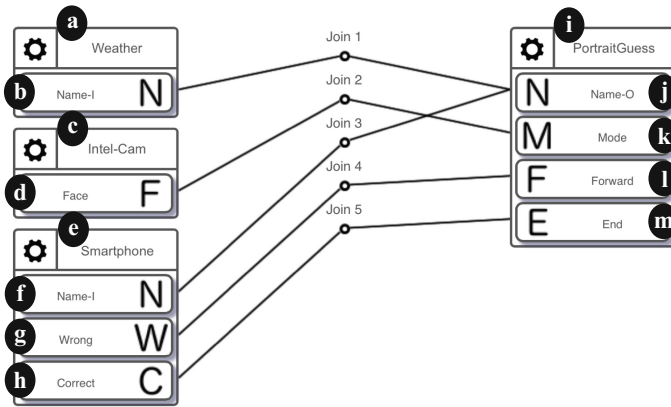


Fig. 10. The IoTalk GUI for a PortraitGuess application.

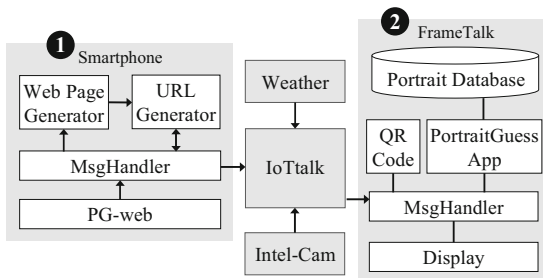


Fig. 11. The functional block diagrams of the PotraitGuess application.



## 5 Conclusion

This paper designed and developed a frame and human interaction platform based on an IoT service platform called IoTtalk. With digital frame hardware, we implemented an IoT device called FrameTalk, which can be connected to an arbitrary smartphone to perform frame art interaction without installing any mobile app. We used two examples to demonstrate how to create FrameTalk applications. The first example is CalligraphyTalk that allows the audience to interact with the frame to write the poems. Based on CalligraphyTalk, an artwork called “When the Cloud Rising” won a Finalists’ Award in Asia Digital Art Award FUKUOKA in 2018 [11]. The second example is PortraitGuess that allows the audience to guess historical figures through the frame. PortraitGuess was one of the 2017 FarEastone IoT Contest award recipients [12]. A demo for CalligraphyTalk can be found in [13].

## References

1. Nixplay. <https://www.nixplay.com>. Accessed 7 May 2018
2. Meural. <https://meural.com>. Accessed 7 May 2018
3. Yiiisu. <http://yiiisu.com>. Accessed 7 May 2018
4. FRAMED. <https://frm.fm>. Accessed 7 May 2018
5. Lin, Y.B., et al.: EasyConnect: a management system for IoT devices and its applications for interactive design and art. *IEEE IoT J.* **2**(6), 551–561 (2015)
6. Lin, Y.B., Lin, Y.W., Huang, C.M., Chih, C.Y., Lin, P.: IoTtalk: a management platform for reconfigurable sensor devices. *IEEE IoT J.* **4**(5), 1552–1562 (2017)
7. Lin, Y.W., Lin, Y.B., Hsiao, C.Y., Wang, Y.Y.: IoTtalk-RC: sensors as universal remote control for aftermarket home appliances. *IEEE IoT J.* **4**(4), 1104–1112 (2017)
8. Processing. <http://www.processing.org>. Accessed 7 May 2018
9. Hsiao, C.Y., Chen, L.K., Lin, Y.B.: Interacting with paintings using IoTtalk. In: *IoT 2017, Proceedings of the Seventh International Conference on the Internet of Things*. ACM, New York (2017). <https://doi.org/10.1145/3131542.3140269>
10. Hsiao, C.Y., Huang, C.C., Lin, Y.B., Lin, Y.W.: Flower sermon: an interactive visual design using IoTtalk. *Mobile Netw. Appl. J.* (2018). <https://doi.org/10.1007/s11036-018-1086-z>
11. Asia Digital Art Award FUKUOKA. <https://adaa.jp/en/winners/winners2017.html>. Accessed 7 May 2018
12. PortraitGuess demo video. <https://www.youtube.com/watch?v=j7V7Xa0hf30&feature=youtu.be>. Accessed 9 Oct 2018
13. CalligraphyTalk demo video. [https://www.youtube.com/watch?time\\_continue=2&v=1KGgCiFVwPK](https://www.youtube.com/watch?time_continue=2&v=1KGgCiFVwPK). Accessed 9 Oct 2018