

Production Site Visualization Using Mobile Apps

Authors: *Yasuhiro Hara** and *Takahiro Kaneko**

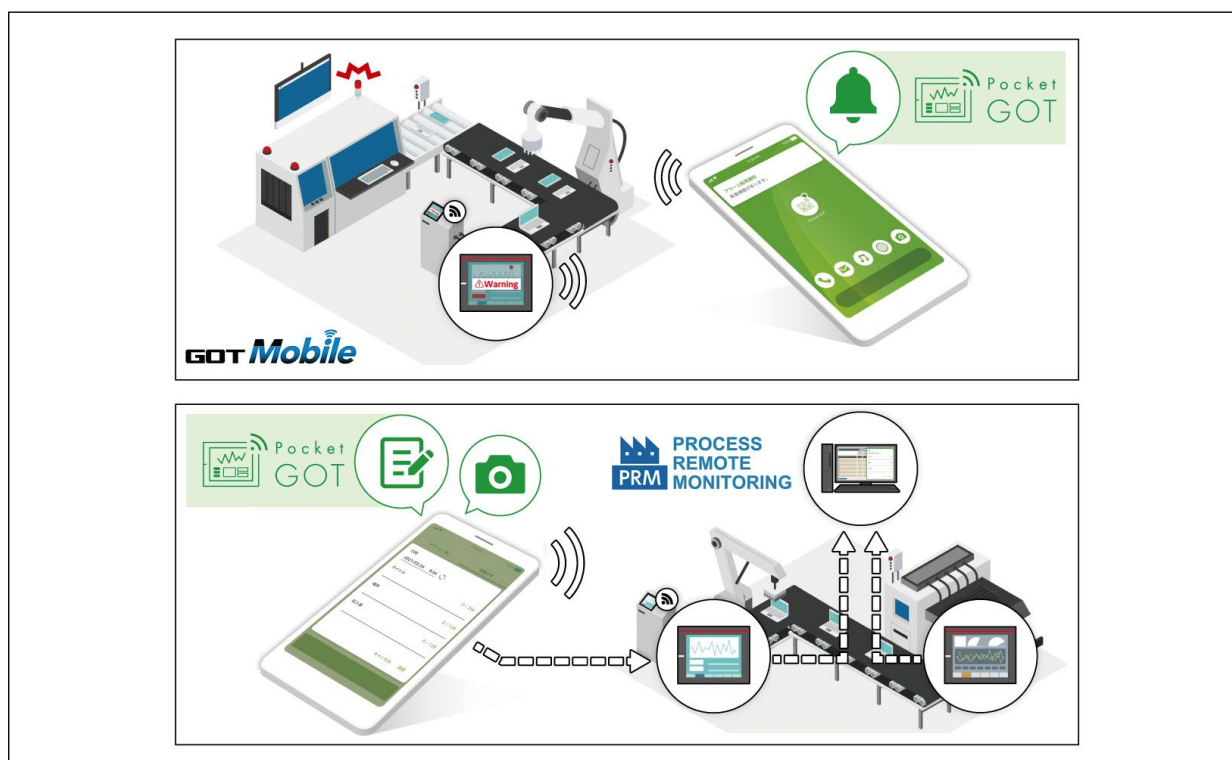
In recent years, the number of devices handled by one operator has increased due to efforts to reduce labor at factory production sites, and it has become rarer for one operator to handle only one device. Furthermore, due to the spread of COVID-19, remote work has become more widespread, and there is growing demand for remote solutions that are easy to introduce at low cost. Meanwhile, when introducing remote solutions utilizing the IoT in equipment, “production site visualization” that collects information from equipment is required.

Accordingly, Mitsubishi Electric Corporation has developed a mobile app “Pocket GOT” linked with “GOT Mobile,” which is a remote solution of Mitsubishi Electric’s “GOT2000 series” touch panel display to assist

working remotely. This has enabled visualization of the production site by using the FA application package “iQ Monozukuri Process Remote Monitoring,”⁽¹⁾ which makes it easy to introduce the IoT in equipment.

The characteristics of Pocket GOT are as follows:

- (1) Receives alarms occurring at the production site and notifies them by vibration, sound, banner, or the like.
- (2) Makes it possible to check information about the site where the alarm is occurring by linking with GOT Mobile.
- (3) Makes it possible to share site information that has been registered as photos, images, and texts on mobile devices.



The Pocket GOT mobile app

This mobile app offers visualization to speed up the on-site response when a problem occurs, by notifying the mobile device of the alarm issued at the remote production site and combining it with the conventional remote solution, GOT Mobile.

1. Introduction

The GOT2000 series (“GOT”) is an industrial touch panel display developed with the concept of “Easy & Flexible (easy to use and highly flexible)” and has been highly acclaimed both in Japan and overseas because it can be connected to various FA devices and can visualize the entire equipment. GOT Mobile (Fig.1), which is Mitsubishi Electric’s remote solution that allows on-site connected equipment to be monitored and operated remotely from PCs and mobile devices such as tablets through the GOT, was well received in the market, leading to demands for greater functionality.

However, GOT Mobile could not quickly grasp the occurrence of problems at the site unless a Web browser was started on a remote PC or mobile device, forcing users to keep a Web browser running at all times.

Therefore, in order to grasp the situation at remote production sites more quickly, in May 2021 Mitsubishi Electric released the mobile app Pocket GOT, which receives notifications of alarms occurring at the production site on a mobile device and is linked with GOT Mobile, which allows equipment to be remotely monitored and operated from mobile devices or PCs.

This paper describes the characteristics of Pocket GOT, and the techniques and ingenious methods used to realize the functions.

2. Characteristics of Pocket GOT

2.1 Remotely grasping problems

Pocket GOT is a free Google Play mobile app for Android*1. Pocket GOT notifies information about alarms occurring at the user sent from the GOT at the remote production site with sound, vibration, or banner on the mobile device. Therefore, unlike in the past, operators can quickly grasp problems at the production site without having to keep a Web browser open on their mobile device (Fig. 2).

2.2 Checking equipment status and alarm history

By linking with Mitsubishi Electric’s remote solution GOT Mobile, Pocket GOT can start the Web browser used by GOT Mobile to display a dedicated screen for GOT Mobile. Therefore, the mobile device makes it possible to check information about where the alarm is occurring at the site and the alarm history (Fig. 3).

2.3 Batch display of working memos

Working memos (daily reports, reports, etc.) with images created on a mobile device with Pocket GOT installed can be sent to GOT. In addition, Pocket GOT makes it possible to easily build a system without specialized knowledge and collect working memos from

the GOT installed on each line at the production site by using Mitsubishi Electric’s FA application package “iQ Monozukuri process remote monitoring,” which allows equipment to be made IoT-enabled. As a result, working memos in ledger format can be displayed simultaneously on PCs in the office, and site information registered in the form of photos, images, and texts can be shared (Fig. 4).

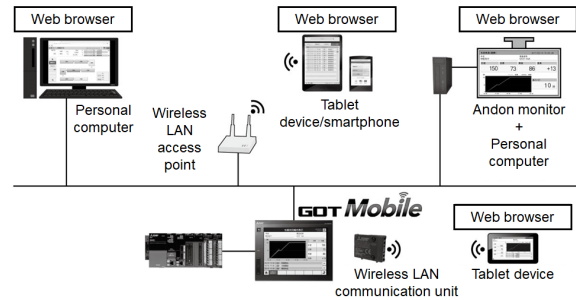


Fig. 1 Configuration of GOT Mobile system

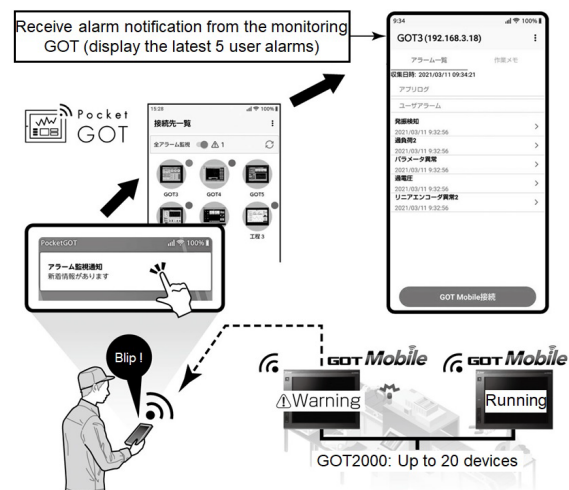


Fig. 2 Remotely grasping problems



Fig. 3 Grasping equipment status and alarm history

*1 Android is a registered trademark of Google LLC.

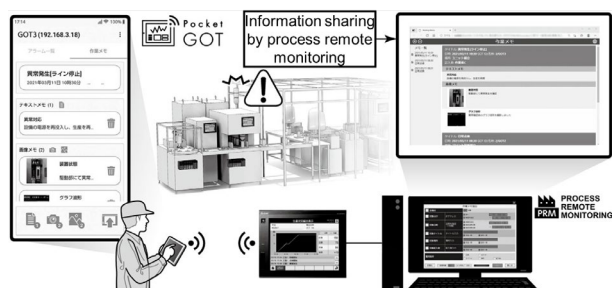


Fig. 4 Batch display of working memos

Table 1 Comparison of mobile app types

Characteristic	Native app	Web app	Hybrid app	Importance
Terminal function restrictions ²	O: Not restricted	x: Restricted	Δ: Partially restricted	↑ High Low
Network connection	O: Not required	x: Required	Δ: Partially required	
System configuration	O: Easy	x: Complicated	x: Complicated	
Operational stability	O: High	x: Dependent on the environment	Δ: Partially dependent on the environment	
Terminal versatility	x: Low	O: High	x: Low	
Program versatility	x: Low	O: High	O: High	

² Camera and communication functions, etc.

3. Development of Pocket GOT

3.1 Types of application

The mobile app comes in three types: a native app that needs to be installed and runs on its own, a Web app that does not need to be installed and runs on a Web browser, and a hybrid app that needs to be installed but which mainly runs as a Web app on a Web browser.

In order to receive the alarms occurring at the production site on the mobile device, Pocket GOT needs a process that always runs even in the background of the mobile device. However, since the Web app and hybrid app that run on the Web browser have some functions that cannot be implemented in the background, Mitsubishi Electric has adopted a native app that has no restrictions on such background functions (Table 1).

3.2 Development environment

Mitsubishi Electric selected Microsoft's cross-platform development environment "Xamarin"⁴ as the starting environment for Pocket GOT because the mobile device on which Pocket GOT is installed must support a multi-platform environment (iOS³, Android) and Mitsubishi Electric can use its knowledge of Visual C#⁴ and Visual Studio⁴ which it has long used in its other products.

4. Techniques and ingenious methods for realizing functions

4.1 System configuration

The system configuration of Pocket GOT is as follows. The Pocket GOT function consists of an alarm notification function (Fig. 5 (1)), which notifies when a problem occurs at a remote production site, and a working memo function (Fig. 5 (2)), which compiles the results of inspections at the production site. The alarm notification function uses HyperText Transfer Protocol (HTTP) communication, which communicates in a server/client format using the same port and sends an alarm request via HTTP using a communication library that has a proven track record in GOT Mobile (Fig. 5 (3)), while the GOT Mobile that receives the request notifies the alarm information via HTTP (Fig. 5 (4)). The working memo function sends data consisting of character strings and images recorded at the production site to the GOT via File Transfer Protocol (FTP) communication (Fig. 5 (5)).

In addition, by installing on PCs the FA application package "iQ Monozukuri process remote monitoring" (Fig. 5 (6)), which collects and visualizes information on multiple devices, the PCs can acquire working memo data from GOT (Fig. 5 (7)) and display it simultaneously in HyperText Markup Language (HTML) format (Fig. 5 (8)).

³ iOS is a registered trademark of Cisco Technology Inc.

⁴ Visual C#, Visual Studio, and Xamarin are registered trademarks of Microsoft Corp.

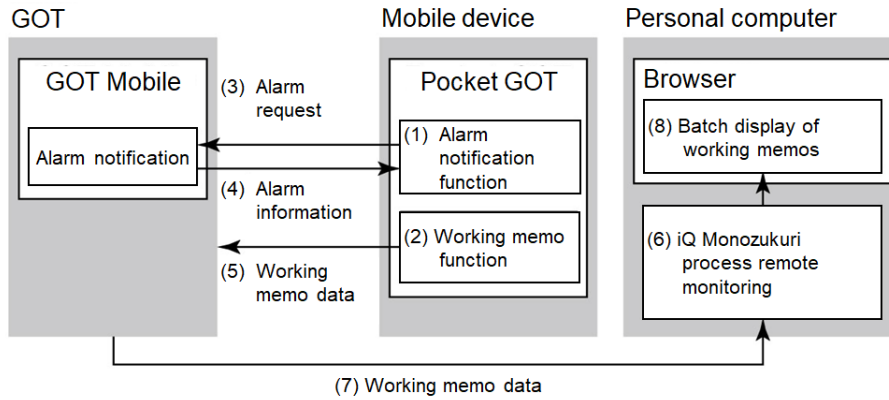


Fig. 5 Configuration of Pocket GOT system

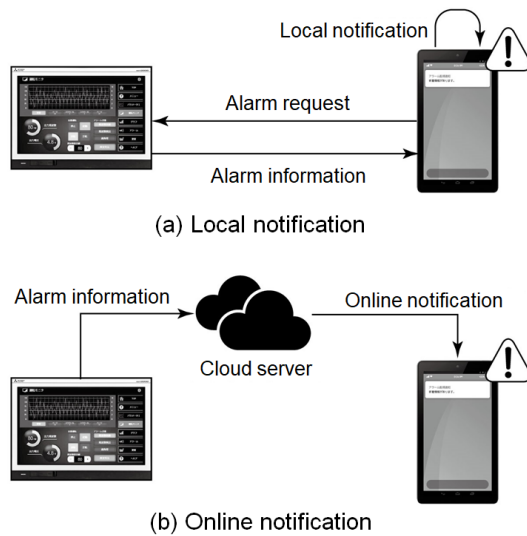


Fig. 6 Local and online notifications

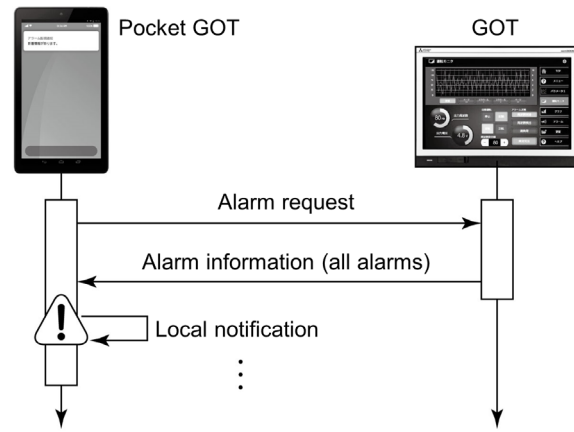


Fig. 7 Sequence of local notification

4.2 Alarm notification method

There are two types of alarm notification method: a local notification method, which displays notifications on the mobile device by issuing notifications within the mobile app, and an online notification method, which receives notifications issued by the cloud server by connecting to the Internet and displays them on the mobile device. The former has the advantage that little of the notification function is implemented on the offline device, but the disadvantage that it is difficult to secure operation in sleep mode. The latter has the advantage that background notification makes the implementation easier, but the disadvantages that notification cannot be implemented on the offline device that is not connected to the Internet, and that a server for notification is needed, which complicates the configuration (Fig. 6).

Pocket GOT uses a method involving local notification that enables notifications to be sent to an offline device and that can be easily introduced. In the

sequence, an alarm occurrence status request is sent to GOT from Pocket GOT, alarm information is received from the GOT, and if there is a new alarm, a notification is generated within Pocket GOT (Fig. 7).

4.3 Splitting communication processing when the mobile device is asleep

When the mobile device is active, it has foreground and background processes. In the foreground process, running of the mobile app is prioritized and is basically unrestricted. In the background process, the mobile app is running but the mobile app screen is not displayed, and the running priority is lower than that of the other apps displayed on the screen, which restricts the running of the mobile app. In addition, when the mobile device is in sleep mode, running of the mobile app is strongly restricted.

Considering the various risks that may occur due to such running restrictions, Pocket GOT has been designed to run in the foreground process even in the background process or in sleep mode. However, a problem became apparent that caused a delay in the process of periodically requesting an alarm to GOT from Pocket GOT while the mobile device is in sleep mode.

Therefore, Pocket GOT has been changed to reduce the processing during sleep and ensure the periodicity by modifying the system to determine, within Pocket GOT, whether notification processing is necessary after Pocket GOT has received only the latest one alarm information returned from GOT in response to an alarm request sent to GOT and compared it with the alarm information received last time. By splitting the communication processing, Mitsubishi Electric has created a mechanism that allows Pocket GOT processing to be executed periodically in the background, including in sleep mode (Fig. 8).

5. Conclusion

This paper described Pocket GOT, a mobile app that enables visualization solutions to be introduced easily and at low cost, thus speeding up on-site response when a problem occurs by linking with the GOT2000 series, GOT Mobile, and iQ Monozukuri process remote monitoring. Mitsubishi Electric will keep improving its user-friendliness and functionality.

Reference

(1) S. Okojima, et al.: FA Application Package “iQ Monozukuri PROCESS REMOTE MONITORING,” Mitsubishi Denki Giho, 94, No. 4, 236–239 (2020)

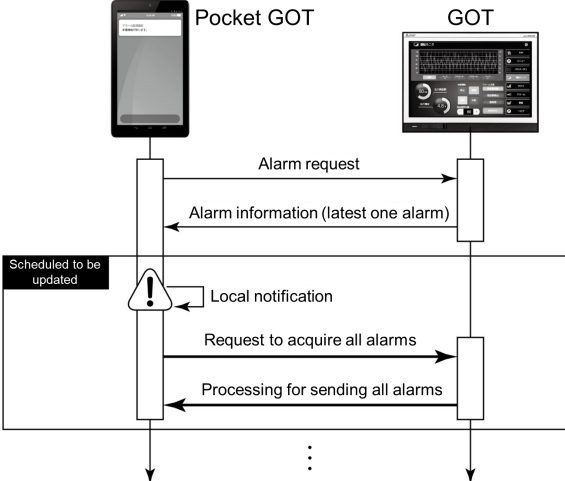


Fig. 8 Splitting communication processing