Journal of Engineering and Applied Sciences 12 (Special Issue 4): 6775-6780, 2017
ISSN: 1816-949X

Process Division Software Design Techniques for Reliability and Continuous Service Management

[1]Woon-Yong Kim and [2]Soon Gohn Kim
[1]Department of Computer and Internet Technique, Gangwon State University,
115 Gyohang-ro, Jumunjin-eup, Gangneung-shi, Gangwon-do, Korea
[2]Department of Computer and Game Science, Joongbu University, 101 Daehakro,
Chubu-meon, Gumsan Gun, Chungnam, South Korea

Abstract: Various service platforms using a variety of Iot technologies have expanded into a wide range of industries and they are demanding a more stable system environment in an independent configuration. In this study, we propose the software management model including the process divide structure and the synchronization techniques in the software design process to make a service platform that is required continuous and stable structure. We apply process divide technique using IPC mechanism in the software design process to take advantage of the stable and independent structure of the core functions. By providing a process divide cooperative structure that is based on the IPC (Inter Process Communication) mechanism in the software design process, we can perform critical process management more effective and independent in the design process. And also it provides more active and ongoing management structure through management model and synchronization techniques. This model will be able to run the software faster and more reliably because it operates independently from the specific process to deal with error conditions that may occur in the unpredictable system environment. In the future, the software design process based on the process divide model will be able to configure a more efficient model by applying to various industries fields.

Key words: Internet of things, inter process communication, process divide, sustainable architecture, software design, mechanism

INTRODUCTION

The various services have been generated by a variety of structures that human interaction is minimized in accordance with the increase of the service platform based on IoT technology (Gubbia *et al*., 2013). In this case, the most important factor is the implementation of reliable service. In other words, it is necessary to cope with problem situations in the most stable platform configuration for a variety of situations and responding to them effectively. In general, a package is composed of a single process. This structure has a weakness for the process will die when a crash occurs in any part in the process. In particular, the combined structure of the various services may be generated a lot of exceptions situation. So, we need to control these problems efficiently. Specifically, this problem in the system where human intervention is minimized exerts serious harm to stability (Schaeffler, 2012; Nahm, 2015).

After all, during the software design, through the separation of the processes in consideration of the importance aspect, the management techniques that manage the functions to be constantly maintained at a stable is needed for coping with the problem. When the system occur a crash from an exception of a specific part of the software, the system will be able to make a more reliable system to process management while maintaining the process that need to stay alive.

Literature review

Processes separation technique: In general if the process is separated and this may provide an important function in a more stable and has a feature that can more effectively manage the available memory in the platform position. Android system provides a variety of service environment for these processes management (Choi, 2012).

By default, the single package is executed in one process. In this case, during a variety of services is running at the same time in one process, if a particular service has an exception, the entire system will be shut down. In addition when all of the service is running in

Corresponding Author: Soon Gohn Kim, Department of Computer and Game Science, Joongbu University, 101 Daehakro, Chubu-meon, Gumsan Gun, Chungnam, South Korea

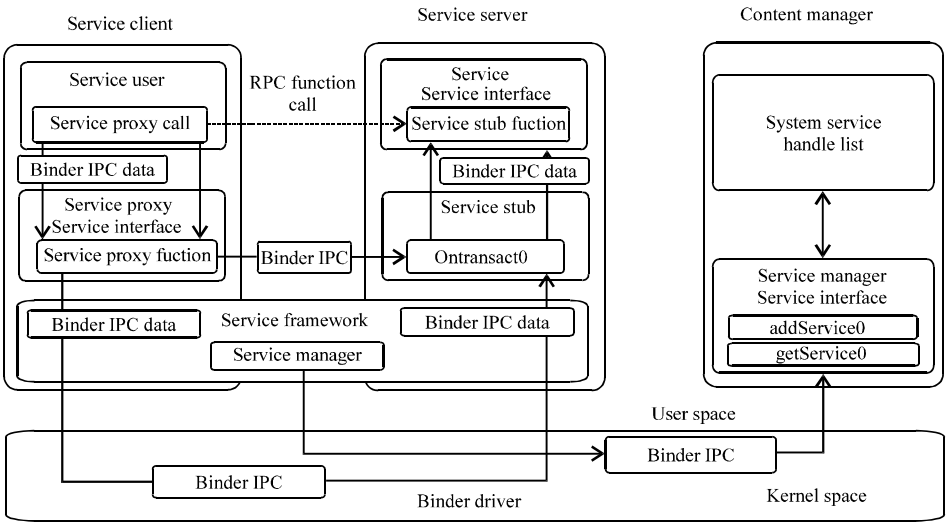

Fig. 1: Inter process communication operation architecture in Android system

one process it occur performance problems. And also using the memory affects the other services when a specific service or views use a lot of memory because the process for the single package is determined the maximum available memory. To solve this problem, there is a need for a process divide structure. This process division model needs to reflect the data sharing method and structure between the different processes in the design process. After all, it is possible to secure a more heap memory when the process is divided, it is possible to provide an important function in a more stable method. In addition, this model provides stability to the background service that is operated a long time and can more effectively manage the available memory in the platform aspect (Samuel and Kovalan, 2016).

Mainly, the background service that operate in a long time or the service that is operated separately from the main process can be separated from main process. This approach reduces the load on the system and also may provide a model that can be stably operated with the exceptions that occur in other services. Typically, when the process is separated, the processes will take each assigned a separated heap space it does not share the memory space to each other. In order to solve this problem, we need to modify an existing business logic with IPC (Inter Process Communication) mechanism. The IPC in the Android system is implemented as a high-performance based RPC (Remote Procedure Call) mechanism that called binder IPC (Gharehchopogh *et al*., 2014). Figure 1 shows the operation architecture for the inter process communication mechanism.

In general, an IPC mechanism is basically the implementation of the RPC mechanism based binder (Binder) and sends and receives data using Kernel space (Anonymous, 2016). At this point, we cannot send or receive an object instance directly. We need to serialize the instance to be transmitted by a separate format to pass the object instance across the process. In the Android system, it defines the interface using AIDL (Android Interface Definition Language) and the activity that received the binder implements RPC (Remote Procedure Call) formant API directly. It can provide functions of various types with synchronous or asynchronous ways. We define API specification using AIDL and translate to the binder interface and stub abstract class. And then, we implement the stub class according to the service bind and unbind cycle. Activity has a structure that can be called a function based on AIDL on the inter process as if using a system service. In this study, we propose a structure that ensures stability of important logic based on the IPC environment and to cope with exceptional situations.

MATERIALS AND METHODS

Concepts of the processes divide model: The process division model comprises some related component in a process and operates these components by each process independently. Figure 2 shows the concept of the processes and software components relationship. In addition, we use IPC Binder for communication between processes. To make such a system structure, we will utilize the Android system using AIDL (Android Interface Description Language) for IPC communication. This architecture has multi process. Some components running on one process and several components than can potentially run on different process.

The concept of the software design to separate processes divides the software functional elements such as main process and service process and then adds the design procedure in considerations of the importance of the each service. It includes the step of designing reliable system architecture. In general, the process should identity the service structure that made in the requirements analysis process and then we assign processes for their structure and set the relationship between each processes. Figure 3 shows the relationship among the processes and service module.

Each process is allocated based on those that are related to services, through the process control module that is main process to control all processes; they handle the life cycle and exceptions. When the problem situation occurs, the module is to maintain the normal flow by the process was killed and then restart that process with recording log information at the same time, so that, we can identify the problems without shutdown of the system and then fix the problems through an upgrade service.

The design of processes division model: In this study, we propose the process division model using the software architecture of the digital signage platform for the lift system that is operated by Kiosk mode. The digital signage platform for the lift is located in the lift inside, lobby and each floor for the display of the lift information and advertisement. And this system allows to efficiently servicing the elevator status information and the advertisement information by itself without user intervention. This system has a feature that runs continuously in the year. Also, has a feature that must support a minimum set of information services in a problem situation. To provide these characteristics, we need to efficiently manage the problem that occurs in a

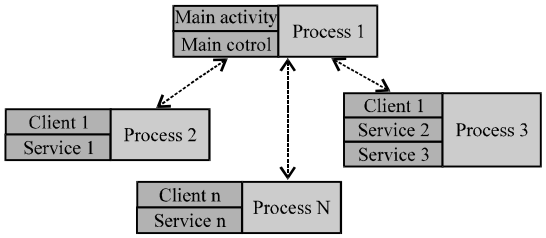

Fig. 2: The concepts of the processes division model to divide software components in a process

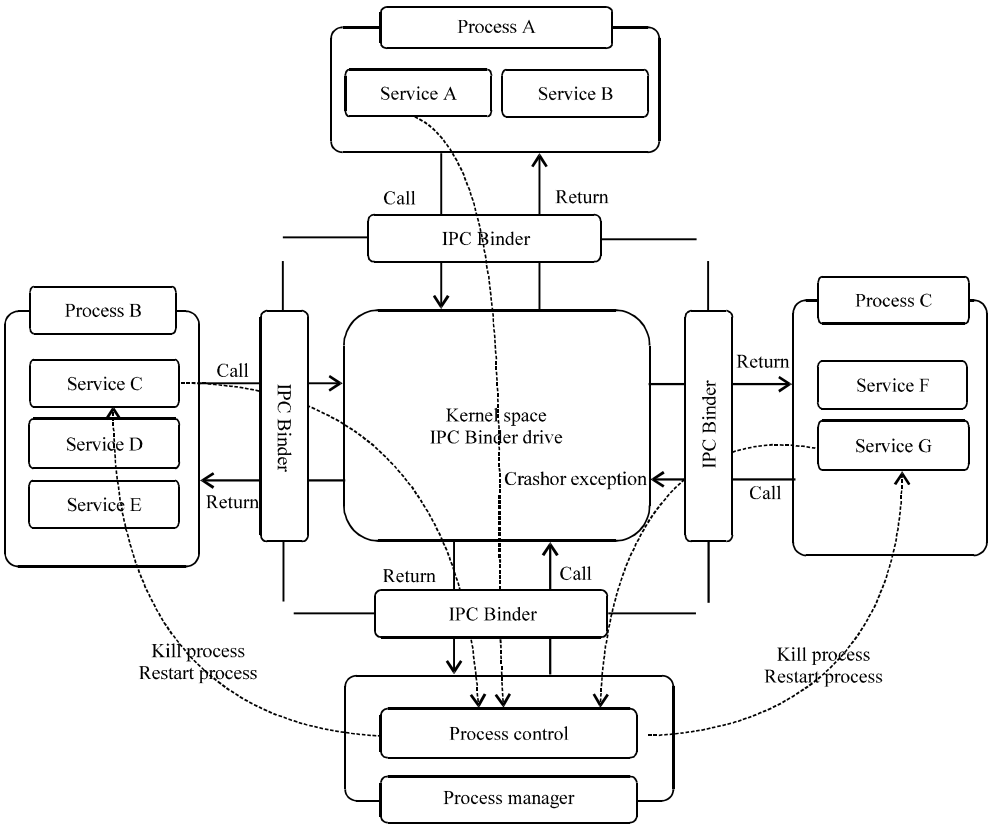

Fig. 3: The relation model of the processes for processes allocation and control flows

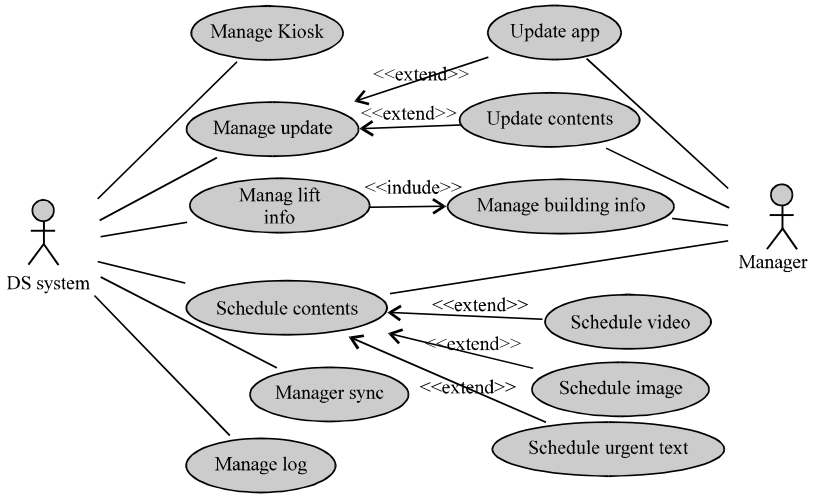

Fig. 4: The requirement specification of the digital signage platform for the lift system based on Kiosk mode

specific service and create a non-stop service structure. The general requirements of the digital signage system of the lift are shown in Fig. 4 and 5.

Kiosk-based Digital Signage Software for a lift has various components such as Kiosk, update, info, contents, sync and media service. These components may be operated independently or in association with each other. When designing the architecture we can define process separation structure in consideration of the importance and independence of the required service configuration. When designing the process separation structure we identify the non-stop service elements, the relationship of the services and memory utilization structures.

We can analyze the associations and relationships for main components through the requirement specification of the lift digital signage platform service.We were separated by base components for driving and the update and schedule management components and media service components here. First, base components including info service for data transfer module, Kiosk service that is an operation configuration and sync service that synchronize among the devices is assigned to one service process. In addition, we have defined the content, update service to support the schedule management and the updating process in one process. Finally, we assigned the media service component into a process to service the media player for a lift. The components are assigned to these processes are operated independently by each process; it is possible to exchange data between them when necessary. In addition, when a specific service process stops this situation is recognized by the process manager service and then the system can provide service without system stop through exception

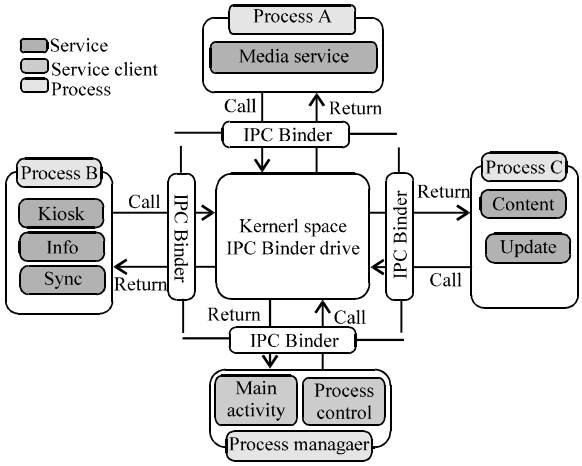

Fig. 5: The processes separation model of the digital signage platform for the lift system based on Kiosk mode

handling. Figure 5 shows the Digital Signage Software process separation model for a lift based on Kiosk service.

The process division model has four processes in a Fig. 5. First process includes Kiosk, info, sync services and second process includes content and update service. And third process has media service. The processes are control by main activity and process control module.

Each process is connected to each other via. the IPC binder and can exchange data with the kernel space. The operation is managed by the process manager process.

RESULTS AND DISCUSSION

Implementation model: Implementation model for digital signage platform configuration for the lift is shown in

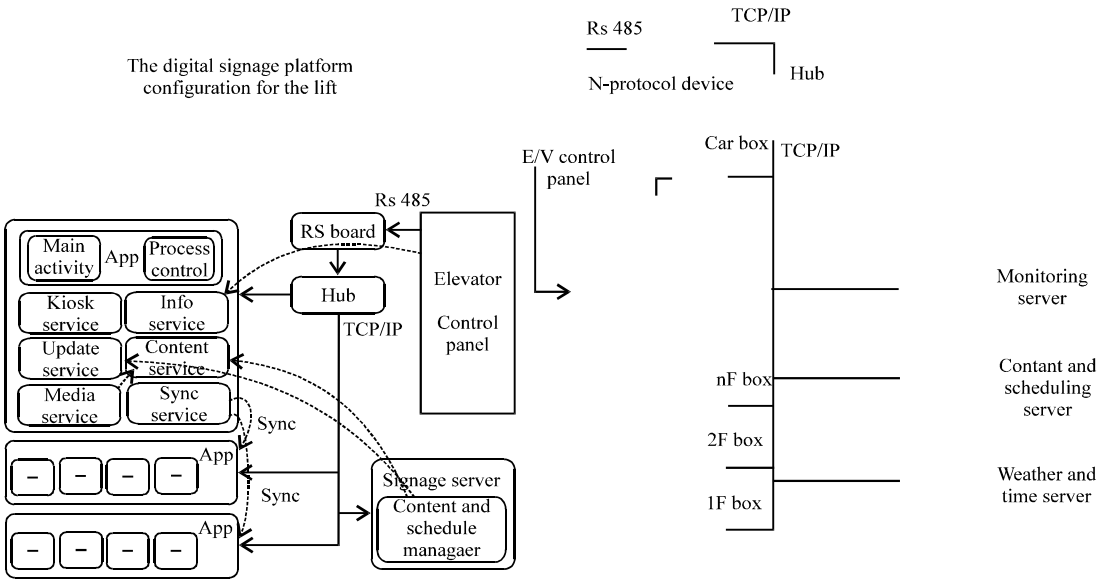

Fig. 6: The processes division model of the digital signage platform for the lift system based on Kiosk mode

Fig. 6. Lift information provided by elevator control panel is converted to TCP/IP communication methods from the system communication method such as RS-485 and then is passed to the main set top box and the monitoring server. And also the content and schedule information provided from content server is passed to the main set top box. In addition, main set top box share the content and information through the synchronization process on the devices displayed on the elevator floor. The set top box applied in this process has a serviceable structure without interruption through the process separation model.

The digital signage platform for a lift consists of an information server for contents management and monitoring information, a module of the control and display the lift status and display devices for displaying information and multimedia service in a lift. Content management and monitoring server is responsible for monitoring the devices, scheduling content, information transmission and update management. Lift controller is providing a lift station status and control environment, the display device for the display of information in the elevator is responsible for effectively displaying and managing all of this information. This display device located inside the lift or each floor is been operating independently without any user interaction. This environment requires uninterrupted service structure without any user intervention. A process division structure presented in this study can be an effective management method when the devices that are minimized user interference are assigned the number of services. The display devices is operated by Android system that is possible to make processes division.

CONCLUSION

Various service platforms for IoT technologies have expanded into a wide range of industries. And the embedded system such as Kiosk and convergence service systems is required of the various needs for building a stable and efficient system environment. In this study, we proposed the processes division software design architecture for continuity and stability service environment. And, we applied process division technique using IPC mechanism in the software design process to take advantage of the stable and independent structure of the core functions.

SUGGESTIONS

For this, we suggest the processes division models based on IPC mechanism for memory management and stability services. And, we proposed the cooperative structure for the separated process model. It provides more active and ongoing management structure through management model and synchronization techniques. This model will be able to run the software faster and more reliably because it operates independently form the specific process to deal with error conditions that may occur in the unpredictable system environment. Based on this, we will be able to build a model that can quickly response to and stably operate the system when it may occur the unexpected situations.

REFERENCES

Anonymous, 2016. Android IPC mechanism. LinkedIn Corporation, Sunnyvale, California, USA. http://www.slideshare.net/jserv/android-ipc-mechanism.

Choi, M., 2012. A Platform-Independent Smartphone Application Development Framework. In: Computer Science and Convergence, Jong-Hyuk, J., J. Park, H.C. Chao, S.M. Obaidat and J. Kim (Eds.). Springer, Dordrecht, Netherlands, ISBN:978-94-007-2791-5, pp: 787-794.

Gharehchopogh, F.S., E. Amini and I. Maleki, 2014. A new approach for inter process communication with hybrid of message passing mechanism and event based software architecture. Indian J. Sci. Technol., 7: 844-852.

Gubbia, J., R. Buyya, S. Marusic and M. Palaniswami, 2013. Internet of Things (IoT): A vision, architectural elements and future directions. Future Generat. Comput. Syst., 29: 1645-1660.

Nahm, E., 2015. Implementation of context-aware digital signage system based on mobile CPU platform. Indian J. Sci. Technol., 8: 1-6.

Samuel, S. and A. Kovalan, 2016. A design level optimization approach for functional paradigm software designs considering low resource devices development. Indian J. Sci. Technol., Vol. 9, 10.17485/ijst/2016/v9i21/95208.

Schaeffler, J., 2012. Digital Signage: Software, Networks, Advertising and Displays; A Primer for Understanding the Business. CRC Press, Boca Raton, Florida, ISBN: 978-0-240-810416-6, Pages: 272.