electronics

MDPI

Article

Object-Oriented RFID with IoT: A Design Concept of Information Systems in Manufacturing

Tamotsu Kamigaki

Faculty of Information Design and Sociology, Hiroshima Kokusai Gakuin University, Hiroshima 739-0321, Japan; kamigaki@hkg.ac.jp; Tel.: +81-82-820-2650

Academic Editors: Michael Sheng and Ali Shemshadi
Received: 27 September 2016; Accepted: 4 February 2017; Published: 8 February 2017

Abstract: The Internet of Things (IoT) has increasingly become important in industry. Connectivity over the internet of not only people but also devices (such as sensors, appliances, machines, robots, and vehicles) is leading to a paradigm shift in manufacturing. The Japanese government recognizes this and has stated that IoT connectivity and the methodologies to exploit it are paramount for Japanese industry. Currently, the necessary changes have been realized in large Japanese companies; however, implementation in smaller companies has been lagging, despite the advantages of introducing IoT technologies, due to the high cost. The objective of this research is to suggest a design concept which combines IoT and object-oriented radio frequency identification (RFID). IoT technology is used for collecting, analyzing, and managing data, and an object-oriented RFID system is used as a control process in manufacturing systems. In previous research, the object-oriented RFID system was shown to provide flexible management through the use of a variety of OORFID tags. The current research extends this by introducing object-oriented RFID into IoT systems to improve the flexibility in the manufacturing systems. For the verification of this concept, an experimental IoT system using object-oriented RFID was designed and implemented.

Keywords: radio frequency identification (RFID); Internet of Things (IoT); object-oriented; information and communication technology

1. Introduction

Around the world, through initiatives such as Industrie 4.0 in Germany, Industrial Internet in the USA, and Made in China 2025, there has been discussion of the directions the next generation of manufacturing will take. Common to all these discussions is the Internet of Things (IoT) where machines, factories, and infrastructure are connected through the internet. The Japanese government as well has recognized the necessity of IoT connectivity and methodologies to exploit it for Japanese industry [1]. At this point in Japan, a great deal of connectivity has been achieved by large companies, although in Japan the term Information and Communication Technology (ICT) has been favored. However, smaller companies have not been able to introduce IoT technology because cost-effective technologies have not been available.

Radio frequency identification (RFID) is a well-known ICT technology that can automatically identify physical objects and people much like a barcode system. A large number of studies have appeared investigating RFID [2], and RFID technologies have become indispensable in manufacturing, especially in production management. The use of RFID provides quick and accurate management of the flow of products [3,4].

In previous research, the concept of object-orientation was applied to RFID and named object-oriented RFID (OORFID) [5]. By introducing the mechanism of inheritance to OORFID [6], various kinds of OORFID tags can be managed with improved flexibility, and the RFID memory is also utilized more efficiently. Inheritance is a core technique in object-orientation and allows the reuse of information. As shown in Figure 1, using the inheritance mechanism, a new object (derived object "Aa") can be easily derived from a base object ("A") and given new features. Similarly, a second derived object ("Ab") can be created with a different set of new features added to the base object. Both objects Aa and Ab inherit common features from the base object.

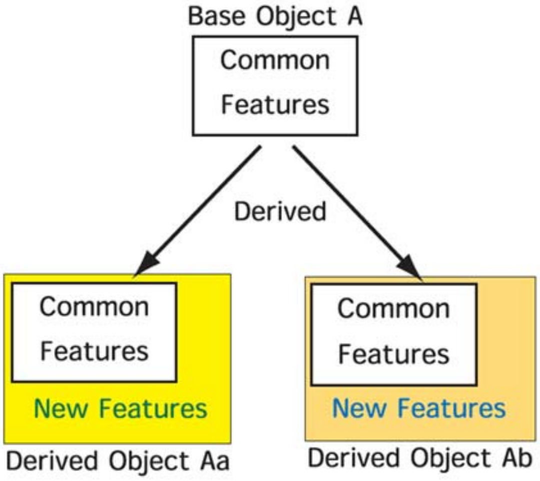


Figure 1. Inheritance mechanism.

In further research, polymorphism was realized using the inheritance mechanism [7]. Polymorphism is an important object-oriented paradigm and is the ability of "one type A to appear as and be used like another type B". Objects of the same class have the same interface and behavior; however, if we use the inheritance mechanism, we can make objects that have the same interface (common feature) but different behavior (new feature) as in Figure 2. To achieve polymorphism, we must make the objects from the base class or from a class that has inherited from the base class.

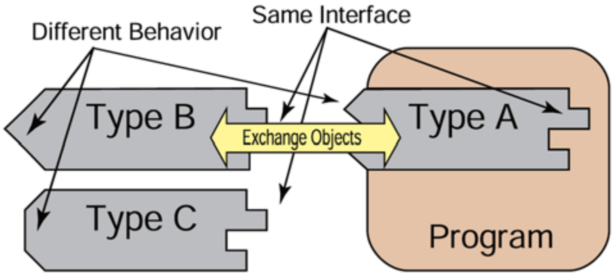


Figure 2. Polymorphism concept.

An RFID tag is usually used as a key to connect to a server, but it also can be treated as an object allowing the application of the mechanisms of inheritance and polymorphism. The objective of this research is to suggest a design concept that combines the IoT and OORFID. The IoT technology is used for collecting, analyzing, and managing data; the OORFID system is used for a control process in the manufacturing systems. In previous research [5–7], it was shown that the OORFID system provides flexible management through the use of a variety of the OORFID tags. The current research attempts to introduce the OORFID into IoT systems to improve flexibility in manufacturing systems.

For the verification of this design concept, an experimental RFID-based manufacturing information system with an IoT element is redesigned. As a proof of concept, a scenario is assumed where mieruka is conducted to observe measurement values obtained from an IoT sensor. Following this scenario, the RFID-based manufacturing information system is redesigned to modify the system using the derived OORFID. As a result, it is shown that the OORFID controls the manufacturing information system—in particular, the communication with the IoT sensor.

2. IoT and OORFID

2.1. Internet of Things in Manufacturing System

The Japanese Government stated in the ICT white paper [8] that IoT means that a multitude of things—not just conventional ICT devices like computers, smartphones, and tablets—will form part of the Internet by virtue of sensors and wireless communications. The IoT concept refers to a state where vehicles, white appliances, robots, facilities, and nearly all other things connect to the Internet and exchange information with each other. According to the concept, as things are converted into data and automation based on these data progresses, new added value will be generated.

In the IoT system, the various things are connected together. However, the connecting itself brings only a few advantages. It is very important how data is collected, processed, and used [9]. The introduction of the IoT to the manufacturing systems has several steps [10] as shown in Figure 3. The first step is the data collection and mieruka. Mieruka is a part of kaizen, and it is the collection of the various data from the IoT devices in order to identify problems and bring them to the foreground. This may use an RFID system to collect data. It also includes a monitoring function. The second one is control. This means controlling devices such as machines and robots through the network. The last one is analysis for automation, which in turn means analyzing the data (sometimes big data) that has been collected to manage the manufacturing process automatically, perhaps by using AI (artificial intelligence). Moreover, the security of information must be considered in the automated control.

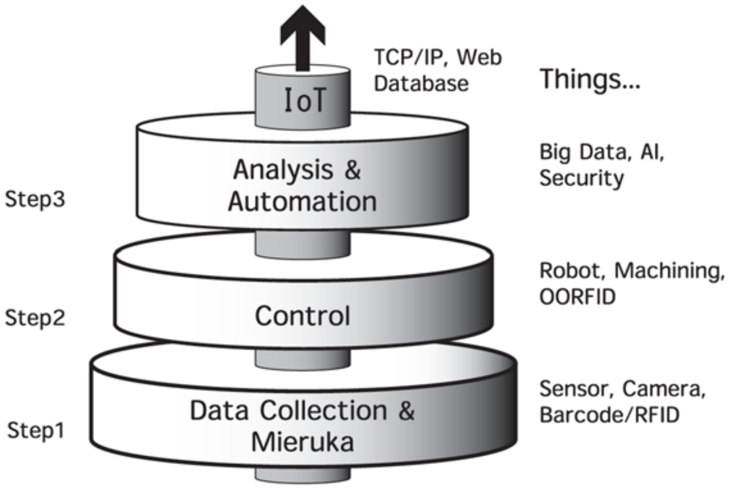


Figure 3. Internet of Things (IoT) Three Steps and Things in Manufacturing.

2.2. OORFID in Contrast with RFID

OORFID is the application of the idea of object orientation to an RFID tag. In the OORFID system, an OORFID tag is assumed as an object. Therefore, the three elements (ID, class information, and definition) are stored in one tag as shown on the right side of Figure 4. The ID is the same as the ID in a typical RFID tag. Class information contains the class name to which the OORFID belongs. The class definition contains fields (object data) and methods (program code for processing data).

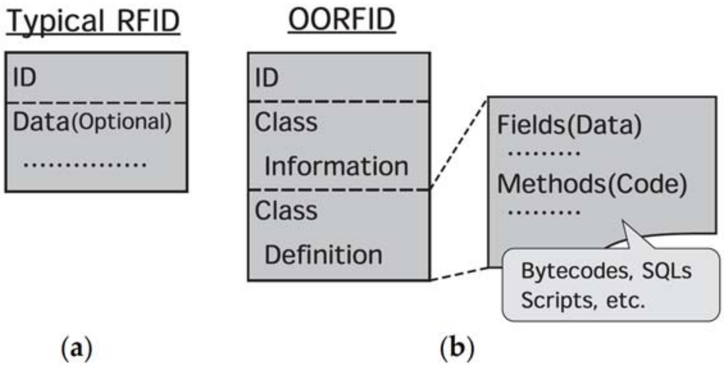


Figure 4. (**a**) Typical radio frequency identification (RFID) and (**b**) object-oriented RFID (OORFID).

Typical RFIDs are used for the management of the products in the same way barcodes are. As shown in Figure 5, an ID in a tag is read by tag readers and is used to access a database on a server computer. Then, the data that is read from the server is transferred through the network system to the machine that uses it [11]. Therefore, the system size tends to be large because we must introduce the tags for each item: the tag readers, the database server computer system with network infrastructure, its software system, and so on.

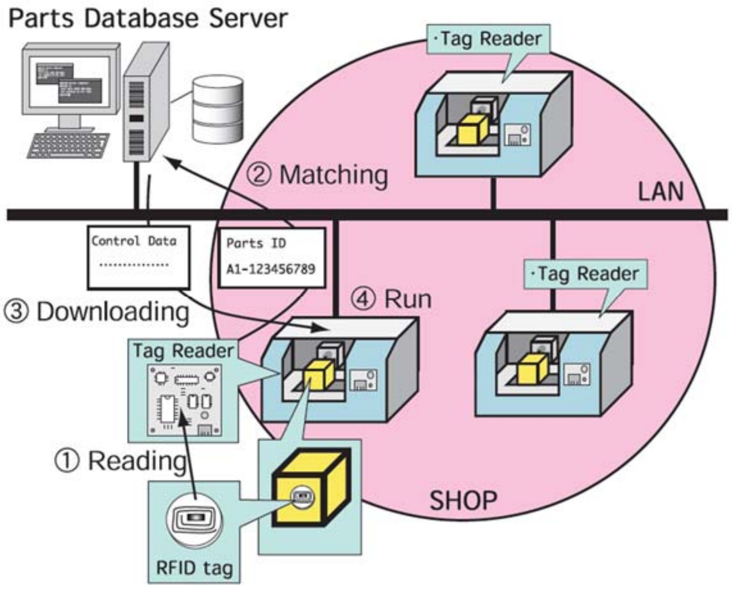


Figure 5. Typical production control data management system using RFID.

Figure 6 shows one informational flow using a typical RFID system. Firstly, a machine reads the ID of an RFID tag and checks the class table to get a class name matching with the ID. Next, using this class name, the machine accesses the parts database server and loads the related data from the server. Then, the machine executes processing with the data. In this kind of system, the parts database must consist of accurate data for all parts. If a change occurs in a part's data, the database has to be modified even if this change is a very small one. In other words, whenever a modification of any part occurs, we must rebuild the parts database.

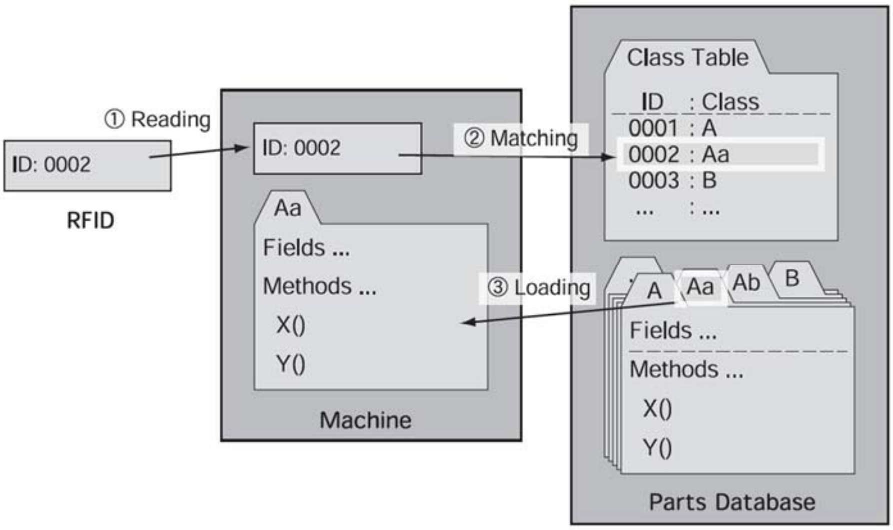


Figure 6. One Informational flow using typical RFID.

Figure 7 shows the informational flow using OORFID [12]. Initially, the database contains the dataset of the parts belonging to the base class. This dataset is almost same as the above RFID system. If a part belongs to one of the base classes, the part data in the database is loaded by matching the class information in the OORFID tag with the class information stored in the database as shown in Figure 7a. The information flow is also almost the same as a typical RFID system.

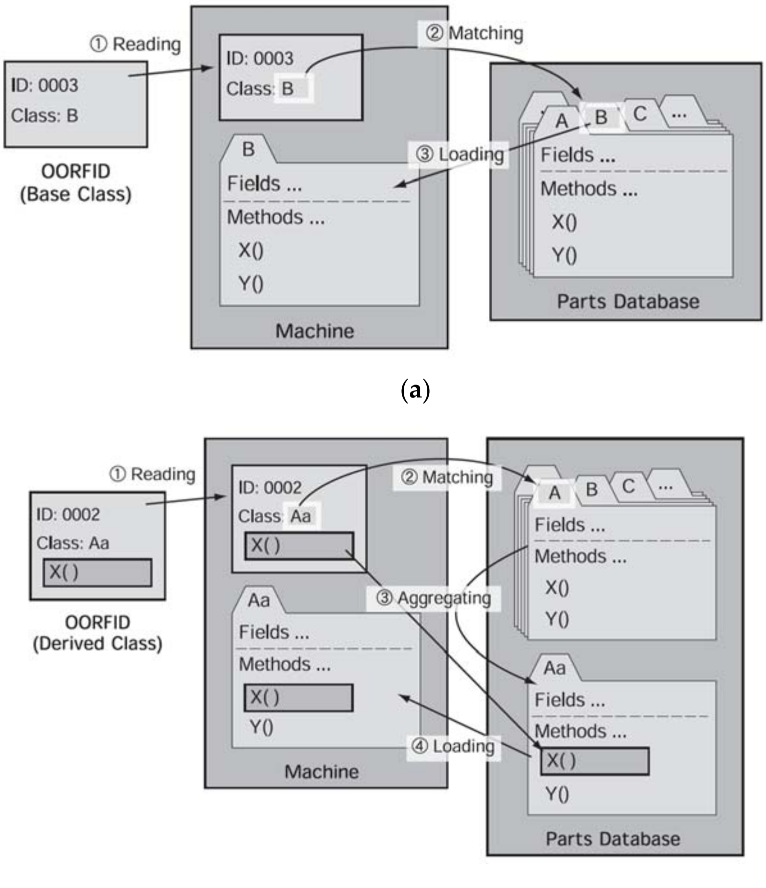


Figure 7. Informational flow using OORFID. (**a**) In case of base class OORFID; (**b**) in case of derived class OORFID.

The OORFID system has an advantage when a modification occurs compared with typical RFID systems. Here, let us assume that there comes a new part that belongs to the class "Aa". The difference between the class "A" and the class "Aa" is only the method "X( )", i.e., the different attribute of these classes is the method "X( )", and the other attributes are same as the class "A". Figure 8 shows the inherited relationship between the base class "A" and its derived class "Aa" in the UML class diagram. In a typical RFID system, it is necessary to update the database, even if a change is small as mentioned above. However in the OORFID system, we create the derived class OORFID that contains the updated method "X( )" as shown in Figure 7b. The derived class OORFID is read and aggregated into the base class "A" that was read from the database, and the updated method "X( )" overrides the method "X( )" inherited from the base class "A". At this point, we can use the new derived class "Aa" to control the machine. In addition, the database can be updated at a later date.

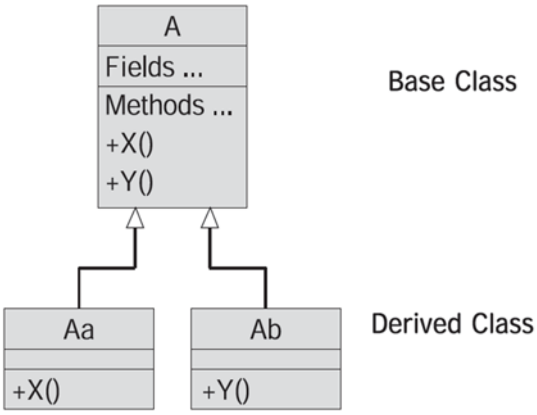


Figure 8. Class Diagram (Derived class "Aa").

Using this mechanism, we can maintain the same interface (an RFID tag), and the different RFID tags bring the different behaviors. In this way, we realize polymorphism.

3. Conceptual Design

3.1. IoT with OORFID and Manufacturing Systems

The ICT systems such as the RFID system are established under the precondition that the system is completely constructed beforehand. For example, all of the production data is stored in the database and there is no modification of it, the infrastructure and the human resources are appropriately assigned, the identification of the parts is tied to the database properly, and so on. In such an environment, the OORFID system is not necessary because a typical RFID or barcode system operates properly to control the process.

However, in actual systems, especially those in the small-sized companies, it is difficult to construct a complete system for some of the following reasons:

- lack of funds for appropriate technologies;
- network is limited to a partial area;
- production data is updated in real-time by upstream demands.

As a result, modifications of the database can occur. In such an environment, the OORFID system can be effective because a system modification is executed as a spot modification of the OORFID.

As mentioned before, the objective of this research is to suggest a design concept that combines the IoT and OORFID. In this system, the IoT technology is used for collecting, analyzing, and managing data; the OORFID system is used as a means to collect data and to control process in the manufacturing systems. Figure 9 is an application of this concept. There is a networked area and non-networked area. In the networked area, all data for production is collected directly through the network. For the operation in the non-networked area, control data is stored in an OORFID as object by rewriting in the writing station before the part comes into the non-networked area. Data collection is executed in the non-networked area to rewrite and update the OORFID with collected data. The data in the OORFID is read in a networked area and transferred into the database. This usage is an application of an OORFID as a storage media.

3.2. Metamorphosis in OORFID System with IoT

Generally, in the software built by a general object-oriented language, once an object is instantiated in the computer memory, it is executed and then disappears from the memory at termination of the program. Therefore, the class has already been written in the software at the development stage and never metamorphoses into another object dynamically.

In the OORFID system, the OORFID tag is handled as an object in the same way as computer software; however, unlike computer software, it may be necessary to metamorphose an object to a derived object as a part moves through the manufacturing process. In a future IoT manufacturing system, if there will be an automation AI and the situation that the AI will decide that an object should be transformed to the derived OORIFD as the result of analysis, the metamorphosis concept of the OORFID will have meaning. This will actually be realized by rewriting the OORFID tag on demand.

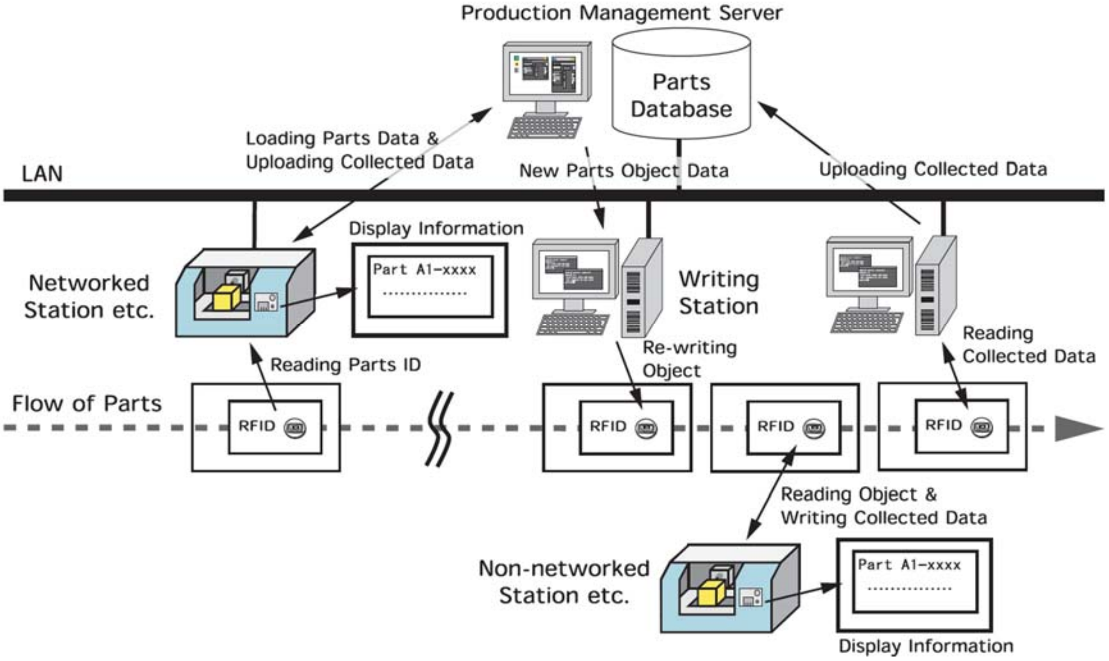


Figure 9. One application of IoT with OORFID.

4. Implementation

4.1. Experimental System

For the verification of the concept mentioned in the preceding chapter, an experimental IoT system was developed using OORFID. The following scenario is assumed:

- The aim is to decrease failure rate of certain parts.
- An RFID system has been previously introduced.
- The failure is suspected to be related to temperature.
- An IoT element is added to the system (a temperature sensor connected to the internet).
- The temperature is observed through the practice of mieruka.

Following this scenario, the RFID-based manufacturing information system is redesigned to allow modification of the system using OORFID.

4.2. Application Design and Implementation

Using the assumed previously developed RFID system, it is replaced with the OORFID system. Firstly, a derived class OORFID has been designed from the base class that was normally used in the original system. The difference between the base class and the derived class is a method used for reading the temperature from the IoT sensor (Figure 10) and displaying the measured value as shown in Figure 11.

Next, the method must be coded. In this system, the method is written in JavaScript and HTML5. In the original system, the RFID tag has only the ID and the type of part. The methods of the base class (base.js: 1595 bytes in this case) are stored in the machine. Figure 12a shows the execution of the original system. This window shows only the part number and type. Figure 13a shows the informational flow of the system. The data stored in the base class OORFID (82 bytes) is simply displayed in the windows. Figures 12b and 13b show the execution in case of the derived class OORFID (595 bytes). This OORFID contains the ID and the archived methods. Once these methods are read and unarchived to the derived class (derived.js: 935 bytes), the same-named methods in the base class are overridden by new methods in the derived OORFID, and these are executed on the web browser to communicate with the IoT temperature sensor and display the temperature of the part.

A PC (Windows 10, PC/SC library, PC/SC compatible RFID Reader/Writer) is used as the machine, an ESP8266 Wi-Fi microcomputer as the IoT sensor, and the MIFARE Classic 4 kB as the OORFID tag. The client program (OORFID reading/writing) and IoT sensor controller are written in C++, the OORFID objects are executed on the NW.js browser.

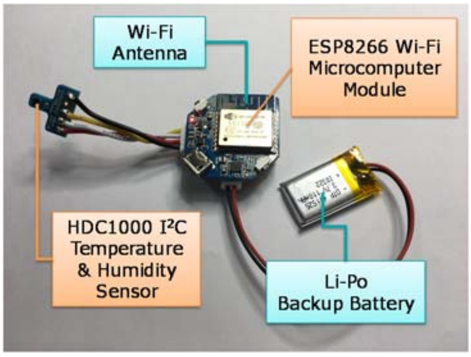
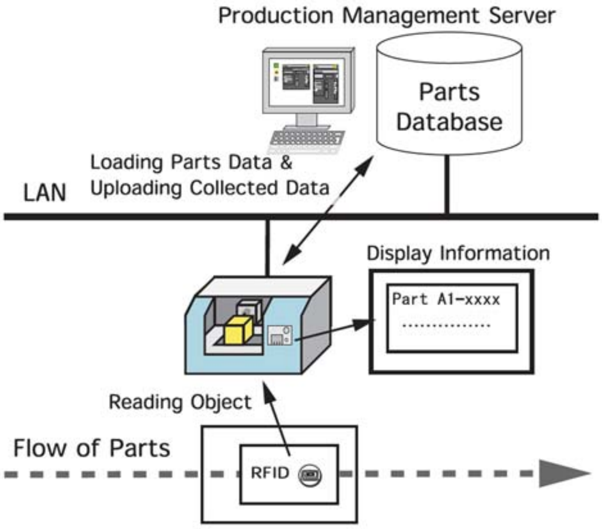
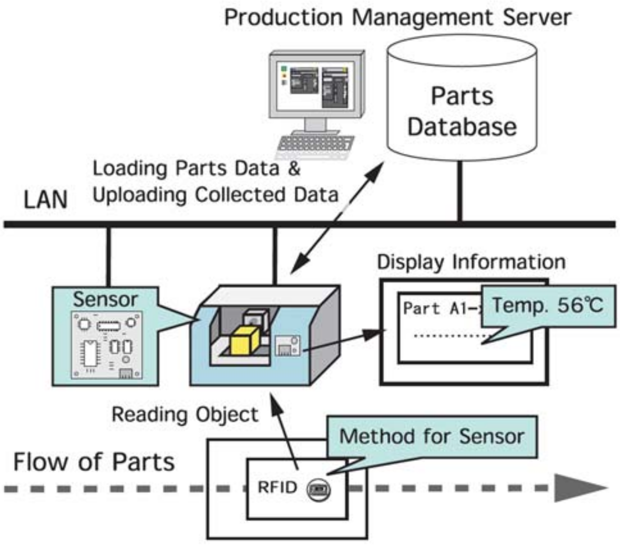

Wi-Fi Antenna
ESP8266 Wi-Fi Microcomputer Module
HDC1000 I²C Temperature & Humidity Sensor
Li-Po Backup Battery

Figure 10. IoT element with temperature sensor.

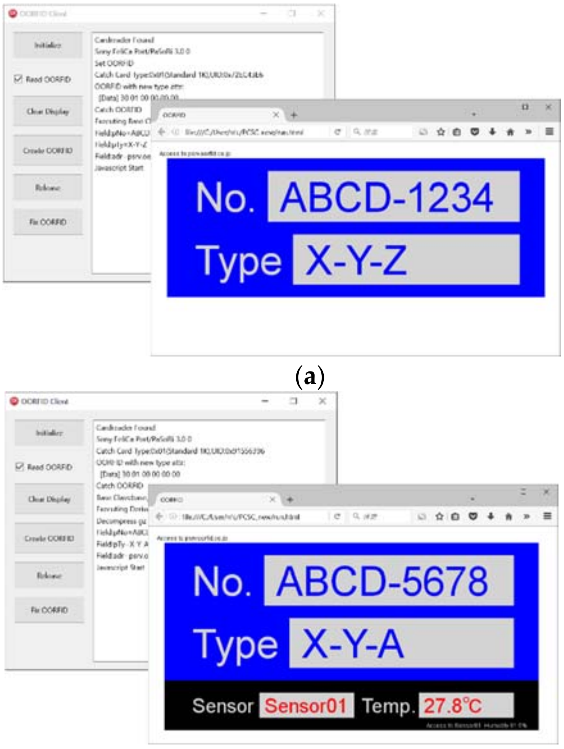


(**a**)



(**b**)

Figure 11. Experimental System. (**a**) Before modification; (**b**) after modification.



```
Cardreader Found
Sony FeliCa Port/PaSoRi 3.0.0
Set OORFID
Catch Card type:0x01(Standard 1K),UID:0x[illegible]
OORFID with new type attr:
 [Data] 30 01 00 00 00 00
Catch OORFID
Formatting Base Cl
FieldpNo=ABCD
FieldpTy=X-Y-Z
Fieldadr=porv.oo
Javascript Start
```

No. ABCD-1234

Type X-Y-Z

(**a**)

```
Cardreader Found
Sony FeliCa Port/PaSoRi 3.0.0
Catch Card Type:0x01(Standard 1K),UID:0x01556[illegible]
OORFID with new type attr:
 [Data] 30 01 00 00 00 00
Catch OORFID
Base Class:base,
Formatting Derive
Decompress gz
FieldpNo=ABC
FieldpTy=X-Y-A
Fieldadr=porv.o
Javascript Start
```

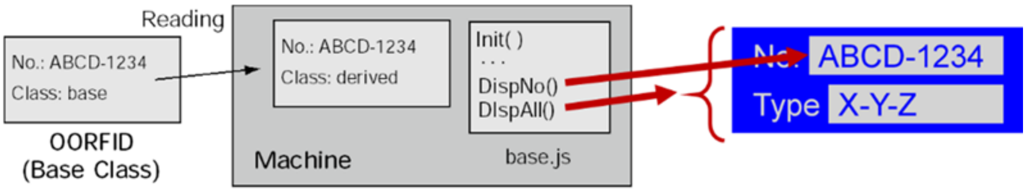
No. ABCD-5678

Type X-Y-A

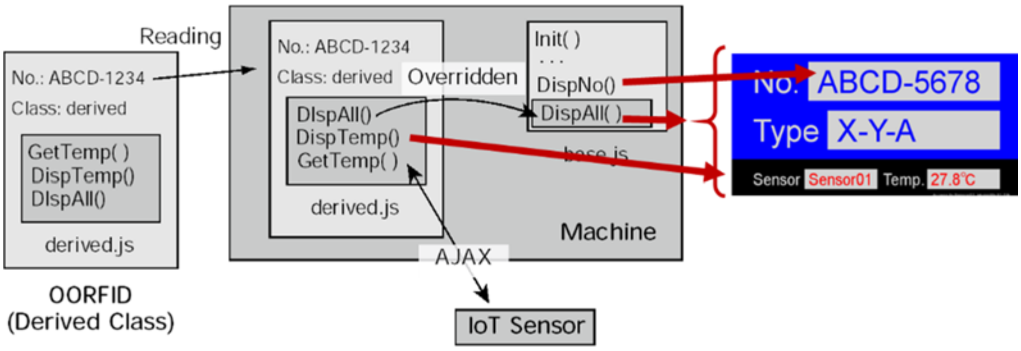
Sensor Sensor01 Temp. 27.8℃

(**b**)

Figure 12. System execution. (**a**) In case of base class; (**b**) in case of derived class.



(**a**)



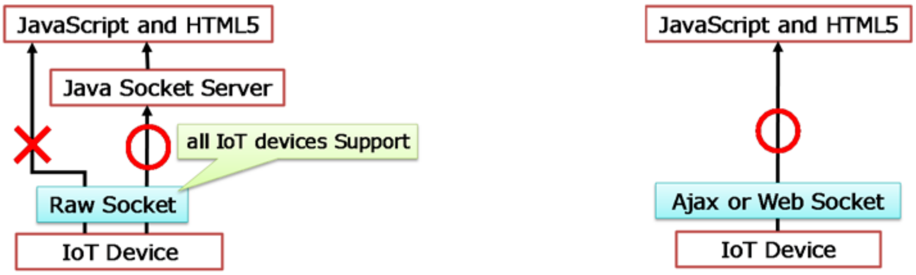
(**b**)

Figure 13. Informational flow of system. (**a**) In case of base class; (**b**) in case of derived class.

5. Discussion

The combination of the IoT and web technologies such as Java or HTML5 makes OORFID possible to execute on mobile devices such as smart phones or tablet computers for the management in manufacturing [13]. This combination enables us to collect and display real-time information for manufacturing by reading OORFIDs with such wireless devices in hand.

To interface with IoT devices, the HTML5 specification provides the WebSocket function. However, currently there are few IoT devices that support WebSocket. Since all IoT devices support a TCP/IP or UDP/IP raw socket control at least for the Internet connection, the initial implementation used the Java socket server to allow communication between the IoT device and the JavaScript and HTML5 programs as shown in Figure 14a. The current implementation uses Ajax (Asynchronous JavaScript + XML) on the IoT device to communicate directly with the JavaScript program as show in Figure 14b. This contributes a resource saving of the methods stored in the derived OORFID tag (e.g., A 1 k Mifare may be used in place of a 4 k Mifare).



(**a**) (**b**)

Figure 14. Communication with IoT device. (**a**) IoT device only supporting raw socket; (**b**) IoT with Ajax or WebSocket.

In addition, in view of software development, writing methods in script program languages such as the JavaScript or HTML5, allows easy and rapid OORFID modification on site when a small change occurs without the need for production management server modification. Therefore, the combination of the IoT and OORFID also contributes to mobility and flexibility.

6. Conclusions

In this research, a conceptual design that introduces OORFID to a manufacturing information system connected to the IoT was described. In this design concept combining the IoT and OORFID, the IoT technology is used for collecting, analyzing, and managing data; and the OORFID system is used for process control to improve the flexibility in the manufacturing systems.

For the verification of this design concept, an experimental RFID-based manufacturing information system with an IoT element was redesigned. As a proof of concept, a scenario was assumed where mieruka is conducted to observe measurement values obtained from an IoT sensor. Following this scenario, the RFID-based manufacturing information system was redesigned to modify the system using a derived OORFID. As a result, it was shown that the OORFID controls the manufacturing information system, in particular the communication with the IoT sensor, and the combination of the IoT and OORFID contributes to mobility and flexibility in software development for the manufacturing information system.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Number 25560017.

Conflicts of Interest: The author declares no conflict of interest.

References

1. *White Paper on Manufacturing Industries (Monodzukuri) 2014*. Available online: http://www.meti.go.jp/report/whitepaper/mono/2014/index.html (accessed on 13 October 2015). (In Japanese)
2. Irania, Z.; Gunasekaranb, A.; Dwivedic, Y.K. Radio frequency identification (RFID): Research trends and framework. *Int. J. Prod. Res.* **2010**, 48, 2485–2511.
3. Watanabe, T. *High-Tech Trend on RFID in United States*; IT Solution frontier: Tokyo, Japan, 2005; pp. 14–15. (In Japanese)
4. NIKKEI RFID Technology and System Design and Development. *All about RFID Applications*; NIKKEI Business Publications: Tokyo, Japan, 2005. (In Japanese)
5. Masuda, D.; Kamigaki, T. Application of Object-Oriented RFID: Data with Processing Code in RFID. In Proceedings of the 8th International Conference on Industrial Management, Beijing, China, 16 November–20 November 2006.
6. Kamigaki, T.; Masuda, D. Designing of Object-Oriented RFID: Introduce of Inheritance Mechanism into OORFID. In Proceedings of the 9th International Conference on Industrial Management, Ostrava, Czech Republic, 26–28 June 2008.
7. Kamigaki, T. Designing of Object-oriented RFID: Implementation of Inheritance Mechanism for Realization of Polymorphism. In Proceedings of the 10th International Conference on Industrial Management, Krackow, Poland, 8–10 October 2010.
8. *2016 White Paper on Information and Communications in Japan.* Available online: http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2016/2016-index.html (accessed on 21 December 2016).
9. NIKKEI Computer. *All about IoT*; NIKKEI Business Publications: Tokyo, Japan, 2016. (In Japanese)
10. IoT Official Approval System Committee; TechFactory, Let's Be an IoT Specialist (1). Available online: http://techfactory.itmedia.co.jp/tf/articles/1604/25/news006.html (accessed on 25 April 2016). (In Japanese)
11. Kamigaki, T. Application of object-oriented RFID: Introduction of RFIDs to partially networked manufacturing systems. In Proceedings of the 11th International Conference on Industrial Management, Venice, Italy, 26–28 September 2012.
12. Kamigaki, T. Designing Object-Oriented RFID and its Application for Manufacturing Systems. *J. Jpn. Soc. Prod. Manag.* **2015**, 22, 27–32. (In Japanese)
13. Kamigaki, T. A Suggestion for Designing Flexible RFID Application Systems. *Autom. Identif.* **2016**, *29*, 33–38. (In Japanese)