

Article

A Web Services, Ontology and Big Data Analysis Technology-Based Cloud Case-Based Reasoning Agent for Energy Conservation of Sustainability Science

Shih-Chin Chen ¹ and Sheng-Yuan Yang ^{2,*}

¹ Department of Fashion Administration and Management, St. John's University, New Taipei City 25135, Taiwan; marschen@mail.sju.edu.tw

² Department of Information and Communication Engineering, St. John's University, New Taipei City 25135, Taiwan

* Correspondence: ysy@mail.sju.edu.tw; Tel.: +886-2-2801-3131 (ext. 6396)

Received: 2 January 2020; Accepted: 14 February 2020; Published: 19 February 2020



Abstract: Energy conservation is one of the important topics for sustainability science, while case-based reasoning is one of the most important techniques for sustainable processing. This study aimed to develop a cloud case-based reasoning agent that integrates multiple intelligent technologies and supports, which can help users to quickly, accurately, and effectively obtain useful cloud energy-saving information in a timely manner for sustainability science. The system was successfully built with the support of Web services technology, ontology, and big data analytics. To set up this energy-saving case-based reasoning agent, this study reviewed the relevant technologies for building a web services platform and explored how to widely integrate and support the cloud interaction of the energy-saving data processing agent via the technologies. In addition to presenting relevant R&D technologies and results in detail, this study carefully conducted performance and learning experiments to prove the system's effectiveness. The results showed that the core technology of the case-based reasoning agent achieved good performance and that the learning effectiveness of the overall system was also great.

Keywords: case-based reasoning agents; web services; ontology; big data analysis; energy conservation; sustainability science

1. Introduction

In this era of information explosion and rapid advancement of network technology, how does one go about finding useful and effective information under the current environment of information explosion? Data mining may be one of the key solutions to this. When combined with appropriate learning techniques, people not only can accumulate experiences at information processing but also can enhance their ability to respond to new information. Case-based reasoning is an important learning technique that solves current problems through previous experiences or past successful cases, which is also one of the most important techniques for sustainable processing. This study thus develops a cloud case-based reasoning agent that integrates multiple intelligent technologies and support systems.

Many studies in the references have proposed intelligent algorithms for environment sensing, which allows users to have a better operational environment of an information system through an environment sensor. The common methods used for environment sensing include the Hidden Markov Model, Artificial Neural Network, Rule-based Reasoning, Case-based Reasoning, and Decision Tree. Smart spaces with environment sensing technology include digital homes, office buildings, laboratories,

etc. The purpose of intelligent environment sensing is to make the environment more comfortable, for example, saving energy and reducing carbon emissions. However, the rules and methods of environment sensing, such as an index applied to parameter settings [1] and differential analysis defined by domain experts [2], require much time for analysis, investigation, and collation when generating new rules. If there is a method to automatically generate rules and its confidence is not lower than the previous two methods, then it can save on time and effort. Therefore, this study is motivated to employ case-based reasoning to analyze the choices and designs of the operations of a back-end support system.

Taiwan is a small island that is densely populated in urban areas but lacks its own resources. Most of the country's resources are supplied by imports at a high proportion of 97%. According to the 2017 Energy Supply and Demand Profile by the Bureau of Energy, Ministry of Economic Affairs, Taiwan's total power generation was 270,278 million kWh in 2017, or 2.33% more than the amount in 2016 (264,130 million kWh); electricity consumption per capita was 11,096.9 kWh, an increase of 2.17% over that in 2016 (10,861.6 kWh) [3]. However, statistics from energy-saving cases assisted by government agencies showed that various fields, such as electricity, lighting, air-conditioning, and office equipment, could help reduce about 20% of potential energy (<https://egov.ftis.org.tw/achievement?cno=4>; accessed on 22 July 2019). So far, most energy-saving systems have focused on how much power can be saved on a single piece of equipment or device. Examples of recommendations include replacing a T8 lamp with a T5 lamp and changing an original air-conditioner to one with an inverter compressor. However, few studies have explored energy efficiency in terms of overall power system operations, which spurs this study to develop a cloud energy-saving information agent system. Moreover, energy conservation is an important issue in sustainable scientific research. To this end, it goes without saying, this study explores the meaning of energy conservation applications.

Smart space encompasses a (1) heterogeneous network, such as Bluetooth, ZigBee, Xbee, and Wi-Fi, and (2) different programming language development platforms, such as Java, C++, C#, and VB. This study took a "back-end intelligent information system" as an example and explored how to integrate different hardware, operating systems, and programming languages, so as to effectively collect, transmit, analyze, and manage related environmental factors [4]. In terms of present web services that are particularly designed for applications and due to the increasing demand for communication between applications, the formulation of uniform standards and protocols for information communication has become more important. The current standards for information communication include HP's e-Speak, Microsoft's .Net strategy, IBM's WSTK (Web Service Toolkit) and WSDE (Web Service Development Environment), and Oracle's Dynamic Service. Sun Microsystems also announced its web services framework, which can be adopted for the J2EE operating platform. The World Wide Web Consortium (W3C) even unified the various standards for web services, including HTTP (HyperText Transfer Protocol), XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), and UDDI (Universal Description, Discovery and Integration). For example, HTTP is a communication protocol widely used on the Internet; XML is a tag language used to describe the interaction between users and service providers; SOAP is a communication protocol based on XML written for the access to web services; WSDL uses XML to describe the details of web services so that applications can easily interact with each other; and UDDI is an XML-based distributed registry of web services of which the goal is to enable web services providers to quickly tell users their available web services. These standards allow web services to have a broader operating base and platform, and that is why this study utilizes web services technology as an important basis to solve the above problems.

Big data means that the amount of data involved is too enormous to be easily retrieved, managed, processed, and organized as information that could be interpreted by humans in a reasonable time period. Hence, big data cannot be processed by most traditional database management systems; rather, it must be performed under software that can run simultaneously on tens, hundreds, or even thousands of servers. Currently, the unit size of a dataset that can be analyzed within a reasonable time

is ExaByte (EB), in which $1 \text{ EB} = 10^{18} = 1000\text{PB}$ (PetaBytes), where $\text{PB} = 1000\text{TB}$ (TeraBytes). Chen [5] mentioned that the size of the dataset does not have to be very large, such as TB, PB, or EB, but it must be a “complete” and “relative” dataset. In other words, the dataset must be able to become a series of useful information after processing and analyzing, in order for an analyst to find out the relationship between information sectors. For this reason, this study adopted a time-series analysis [6] as the basis for retrieving relevant cases. How does one carry out unimaginable and incomprehensible big data analytics? The fundamental solution is to “make it simple.” People can summarize the results of big data analytics and take corresponding actions or make optimal decisions. The biggest challenge of this study is to explore a back-end system based on big data analytics.

Some domestic and foreign studies in the references have used both case-based reasoning and learning to develop relevant information systems. For example, Yu [7] used the case-based reasoning technique to explore implementation methods for an optimal milling machine system that meets the processing needs. Lou [8] combined a case-based reasoning module with a practical database to find the related technologies of knowledge management and development framework for financing, credit, and default cases in the banking industry. Huang [9] noted the key factors for sales personnel to sell sports shoes and established the weights of these factors according to case-based reasoning, so as to get the relevant technology for sales forecasts of new products before these products are launched. Relich and Pawlewski [10] applied case-based reasoning to neural networks to estimate the unique product development costs of a manufacturer. Faia et al. [11] used case-based reasoning, expert system, and collective intelligence to discuss issues related to energy reduction for building energy management systems. Aljuboori et al. [12] proposed a case-based reasoning technique based on association rules to improve the performance of similarity-based retrieval and classed frequent pattern trees (FP-CAR) algorithms and to explore the technology and application of eliminating errors or ambiguous retrievals by case-based reasoning. Chung et al. [13] developed a programming learning and diagnosis system built on case-based reasoning, which can provide feedback and suggestions to learners in a timely manner.

Most of the references mentioned above combined a traditional case-based reasoning technique with corresponding quantitative similarity methods and integrated the combination with the new notions that scholars have improved or developed on their own, so as to highlight the importance of self-developed comparison mechanisms. This study also developed its own quantitative similarity method, by calculating the difference of the time-series data of web services provided by WIAS (Web-service-based Information Agent System) [14] and the corresponding data in the Case Base system [15] and then reversing the difference to determine the similarity of the case. This study easily completed case-based reasoning and learning mechanisms through web services, such as case retrieval, reuse, revision, and preservation, as provided by WIAS. In this way, this study not only proved that the self-developed quantitative similarity method is more suitable for applications but also highlighted that the system operation herein can rapidly reach cloud information services through the characteristics of interface integration and instant amplification of web services.

This system was built on the web services-based information agent system developed by the Intelligent Systems Laboratory of the Department of Information and Communication Engineering, St. John’s University [14]. It used SQL (Structured Query Language) to access the templates and then constructed a common SQL integrated circuit (IC). The function of this system is like an IC. It can connect different parameters and communicate with the corresponding databases and web services, such as OntoDMA (Ontological Data Mining Agent) [14], OntoCBRA (Ontological Case-Based Reasoning Agent) [15], Ubi-IA (Ubiquitous Interface Agent) [16] and shared control so that the corresponding query results can be easily accessed and the relevant information access services can be provided, as shown in Figure 1. The design philosophy of this system was the service-oriented architecture of web services technology, meaning that it took the concepts of SQL IC to construct a cloud database manipulation interface like data storage to quickly and effectively integrate the back-end of the system. For example, it predefines the raw data of energy savings and forecasts its rules and cases. Such a

design maintains the advantages that cloud data can be accessed at any time and retains the consistency of the interface of data storage. Figure 2 shows the flow chart of the WIAS operation. The system hierarchy is Data Layer, Control Layer, and Service Layer [14], from bottom to top.

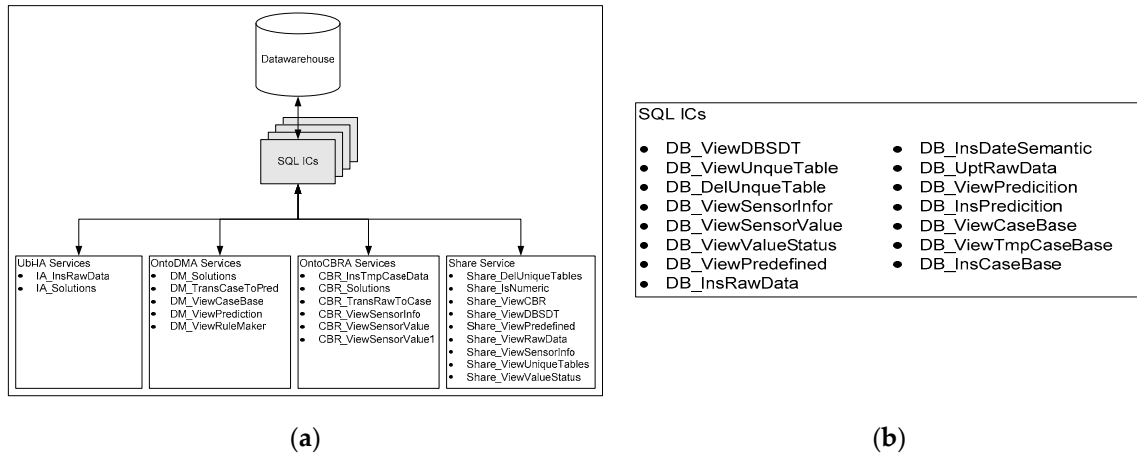


Figure 1. Conceptual framework diagram of Web-service-based Information Agent System (WIAS) operation. (a) Hierarchy chart of web services; (b) Structured Query Language (SQL) integrated circuit (IC).

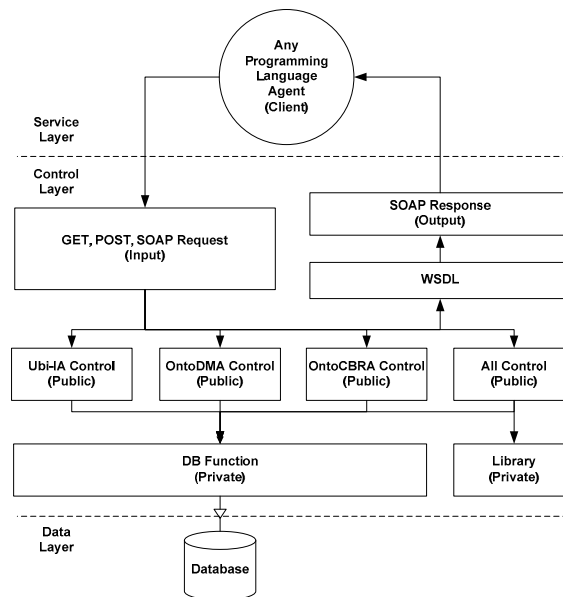


Figure 2. Flow chart of WIAS operation.

In short, energy conservation is one of the important topics for sustainability science, while case-based reasoning is one of the most important techniques for sustainable processing. This study focused on developing a cloud case-based reasoning agent built on web services, ontology, and big data analytics [15,17]. This study first introduced the technology related to the construction of web services platforms and then explored how to use this technology to widely and seamlessly integrate and support the cloud interaction between the back-end information multi-agent systems. In order to confirm the feasibility of the system architecture, this study took an energy-saving system as an example to present relevant R&D results in greater detail. The performance and learning effectiveness of this system was highly verified through initial R&D and a comparison of the system screen.

2. Background and Technology

Ontology was originally a term in the field of philosophy as a statement that explores the nature of the knowledge of life or real things and provides a complete semantic model with the characteristics of sharing and reuse. Figure 3 presents the part of the ontology of energy-saving information, which mainly defines the basic knowledge, hierarchical relationship, and characteristics of various energy-saving equipment and uses these definitions to support the overall operation of the back-end information agent system. We have also quoted from Protégé (<http://protege.stanford.edu/>) as the foundation for the ontology-based cloud energy-saving information system architecture of St. John's University. The implementation of these ontology-based services included converting the semantic distance of the retrieved words and transforming the retrieved words into the corresponding hypernym, hyponym, synonym, and antonym. Moreover, it took WordNet (<http://wordnet.princeton.edu/>) as the base of the comparison model and combined with the Academia Sinica Bilingual Ontological Wordnet (Sinica BOW, <http://bow.sinica.edu.tw/>) to explore the conversion between Chinese information and English information, the link between language information and conceptual architecture, the distinction of the meaning between words and terms, and the connection and usage of the meaning between words and terms. The above-mentioned was the basic structure of the knowledge of this system. Furthermore, this study used Jaccard similarity [17] to estimate the consistency between ontological concepts, using the consistency between the concept of the retrieved words and the corresponding concept of WordNet and its related position to index the domain concept. Finally, it adopted the identification code Synset_ID in WordNet to access the domain concept and support the overall system operation. WordNet is a general online English database developed by the Concepts and Cognition Laboratory of Princeton University in the United States that contains 42,000 concepts. Particularly, this database can list the synonyms and antonyms of words, identify the hypernym–hyponym relationship, and most importantly, enable users to access the WordNet database by SQL and JWNL (Java WordNet Library, <http://jwordnet.sourceforge.net/handbook.html>). This was the main reason for us to take the SQL database to develop an ontology- and Java-based system.

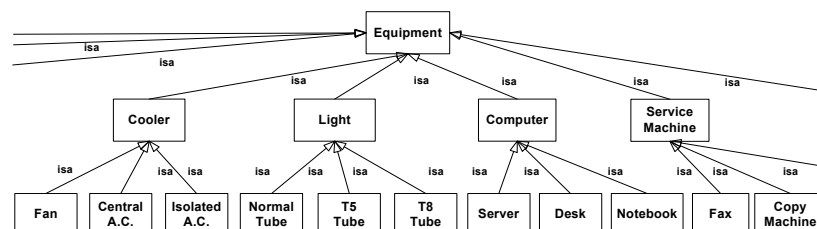


Figure 3. Part of energy-saving ontology.

This study also proposed a parallel reduction mechanism [17] based on big data analytics and divided it into four steps: (1) Generate Preprocess works for the keyword sets corresponding to individual websites; (2) Apply domain ontology and Jaccard dissimilarity to get Maps representing three keyword sets on individual websites; (3) Sort Shuffle corresponding to the optimal three keyword sets on individual websites; and (4) Use the average output of Jaccard dissimilarity and the closest three corresponding keywords that users have inquired to conduct the works of Reduce, as shown in Figure 4.

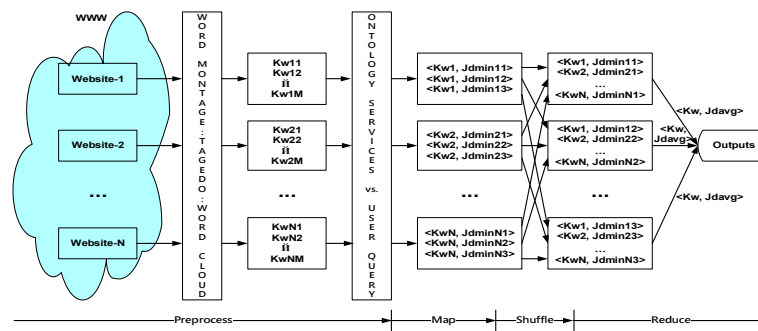


Figure 4. Proposed parallel reduction mechanism.

Following the above literature, this study took the semi-open source framework Hadoop (such as Dropbox) as the context and explored the concept of “R + Hadoop = Big Data Analytics” [18]. This study also built the above-mentioned MapReduce parallel reduction mechanism, integrated the computing between the keywords of domain ontology services support and the corresponding Jaccard dissimilarity, and then supported the various information services of this system based on the WIAS big data analytics technique. The actual operation is shown in Figure 5, which shows the OutputFormat at the end of the domain ontology service index. From the establishment of Jaccard dissimilarity and the ontological databases (OD) of the real data, the above-mentioned concept was able to realize that the big data analytics of the domain ontology index can support the subsequent system operation.

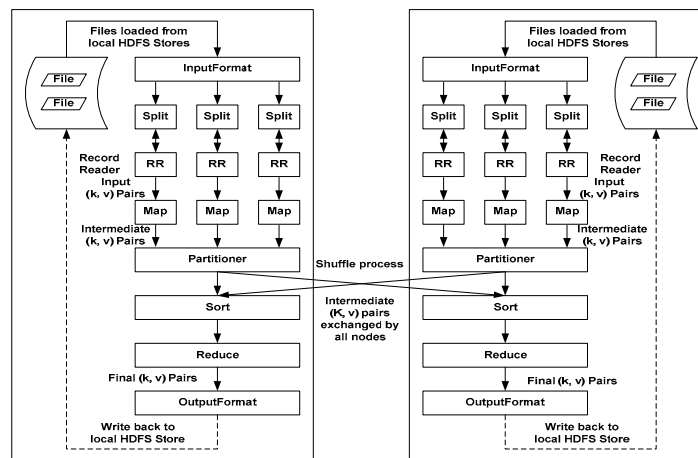


Figure 5. Actual operation diagram of MapReduce.

Figure 6 presents the complete framework of this WIAS system [14]. Users can use various Internet channels to transfer relevant information from Ubi-IA [16] to the back-end information system, while this system can play as a control center that answers the questions of energy-saving information by Solution Finder. Ubi-IA is responsible for providing the processing and conversion of cloud query information and for making intelligent query decisions. The so-called “intelligent query decisions” provide the cloud solutions corresponding to cloud query. The set-up covers a three-stage intelligent query decision process as follows: (1) Process the query one-by-one through OntoDMA to determine whether there is a cloud forecasting solution; (2) use OntoCBRA to conduct a batch of sequence query processing to determine whether there is a CBR solution; (3) search for external solutions from the Internet via OntoIAS (Ontological Information Agent Shell) [19], which originates from Ubi-IA of this WIAS system, and process the query one-by-one to determine whether there is a forecasting solution in accordance with the default rules constructed by domain experts. The database of this system came from the ontology database OD processed by the parallel reduction mechanism mentioned above, as well as the user-related database supporting the information sharing recommendation.

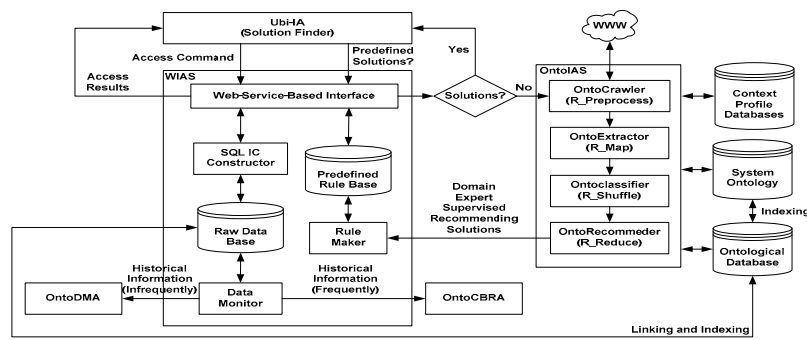


Figure 6. Framework of the WIAS System.

At the beginning of the operation, this system combined the cloud information ontology constructed by domain experts with the corresponding default rules, retrieved the useful information of the words that came out when querying information, and initialized the query information to support the confidence of default rules. At the same time, a similarity algorithm was conducted for the cases corresponding to the WIAS web services support in the cloud information ontology to complete the initialization of the system operation. Once the system started to operate, the system would periodically make a response according to the frequency of querying information [6]. This system employed time-series analysis to calculate the most frequently occurring and the least frequently occurring query information. OntoCBRA generated relevant information of the cases according to the calculation result and matched it with the two-stage time-series forecasting algorithm to trigger the default rules corresponding to OntoDMA adaptation. The domain ontology helped to compare and retrieve the appropriate correspondence in order to effectively add value to the quality of cloud information consultation and sharing, thus enhancing the correctness, authenticity, and integrity of said information. If both OntoCBRA and OntoDMA are unable to appropriately answer the cloud query, then this system will trigger OntoIAS to directly conduct the Preprocess, Map, Shuffle, and Reduce operations of the aforementioned parallel reduction mechanism to complete data research, data retrieval, data classification, and data presentation (or sorting). This system sought out solutions of cloud information from the Internet and added the default rules by domain experts, so as to fully construct the learning cycle in response to query information. Through the three-stage intelligent decisions (OntoDMA, OntoCBRA, and OntoIAS), the purpose of this study—to find a solution for optimal energy-saving information solution—was achieved. This study focused on the development of OntoCBRA under the support of the aforementioned technologies.

3. System Architecture

According to the results of the previous system [15], the main CBR system architecture and its four main steps are unchanged, but two optimization steps for calculating the similarity of the cases are added to more clearly illustrate the relevant calculation procedures, which are illustrated later. Figure 7 presents the framework of this OntoCBRA system. Here, Ontological Database Access Cases (ODAC) are mainly periodically generated by the Historical Information of WIAS and the case generator under the ontology database index of the system. If the directive is whether there is a CBR solution, then the function Case Retrieval will launch ODAC under the ontology database index to accurately and effectively retrieve the same or similar cases. These cases have their similarity calculated based on the retrieved cases. Identical cases will be directly returned to the Ubi-IA; those that are not identical, but meet the standard of trimming (the threshold set by this system), will pass onward to case trimming for further processing. After undergoing trimming, the most appropriate solution comes out and is transferred to the interface agent system for reference. If the user is satisfied with the solution, then this solution will enter into the step of case retainer; otherwise, it will be discarded directly. Finally, the system decides whether the trimmed case should be stored based on case similarity

(beyond the threshold set by this system) and then restores it to ODAC under the system ontology index. In addition, ODAC relies on a case monitor to provide case information, which can become the materials for OntoDMA to establish prediction rules [14].

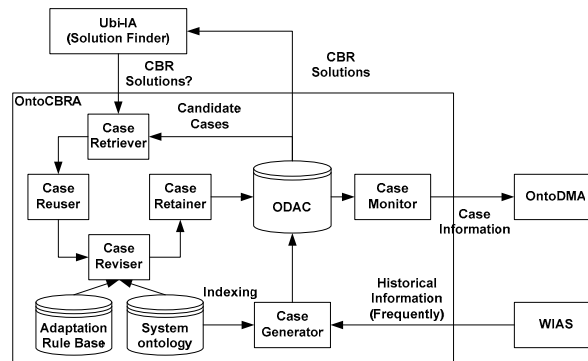


Figure 7. Framework of the proposed system.

The source of the training materials for this case-based reasoning was the sensing data of electrical devices operating in a specific energy-saving monitoring environment and was sent in relay transmission. These sensing data was transmitted to Interface Agent at regular intervals, and the received data was stored in the cloud database via the services offered by WIAS. The cases were generated by the Case Monitor of WIAS by transforming the huge original data into information with semantic meaning. At the end, this information was summarized into cases in the case base, which were generated by the accumulation of transformational processing experiences. In other words, the knowledge content of the cases in the energy-saving information system was planned to that of a description that converts raw data to a semantic meaning for a specific time, area type, area size, measurement and control (MAC: Media Access Control Address) type and sensing data. As mentioned earlier, the case-based reasoning is a technique based on past experiences and successful cases to solve a current problem. Therefore, this study defined a case as the most common information in a given period and its corresponding energy-saving operational mode. In other words, it is the most stable energy-saving solution in the monitoring environment. Figure 8 shows the concept of case generation, for which its trigger was the case monitor of OntoCBRA, which can communicate with WIAS. Its periodic off-line operation can send historical information of the cloud database to OntoCBRA as the material for case generation. However, the original case shown in Figure 8 cannot directly be used in various energy-saving operational modes. Hence, it was necessary to convert the case into an appropriate semantic case in accordance with the corresponding sensor category table, so as to apply it to most energy-saving case-based reasoning mechanisms. The practice example of a case generation is shown as Figure 9 [20]. Figure 9 presents a complete case content, including the start day and time, end day and time, area type name, area size, measurement and control (MAC) type name, and sensing data. This complete case content was divided into three parts: (1) Convert the time period into a semantic meaning; that is, let users understand the meaning of system processing; (2) acquire environmental information and use it to quickly classify or filter a relevant system case to be a solution; and (3) obtain knowledge answers.

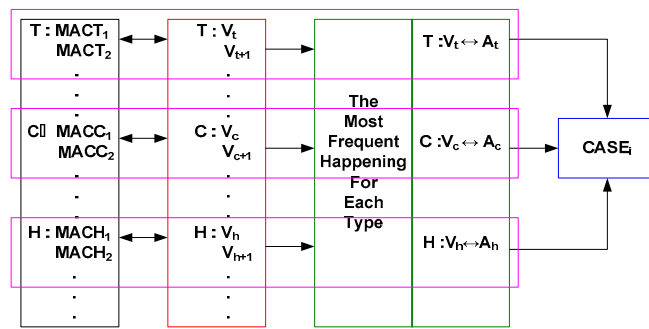


Figure 8. Case generation concepts (T: Temperature, C: CO₂, and H: Humidity).

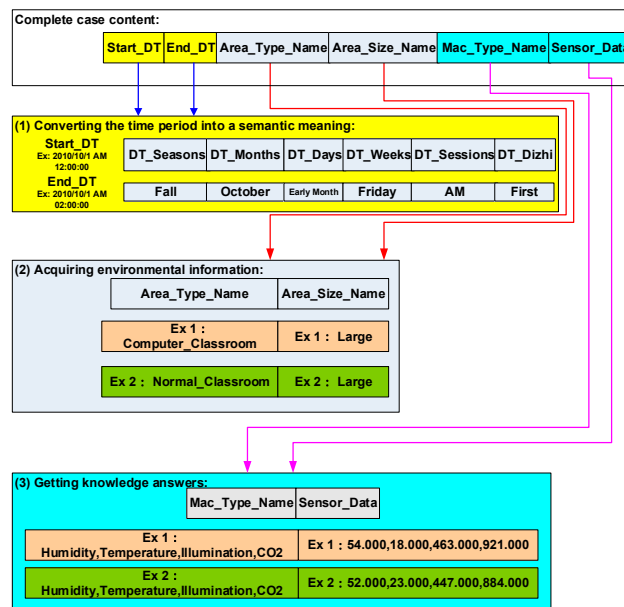


Figure 9. Practice example of a case generation.

Case Retrieval was divided into two stages. Stage 1 was the semantic time period and location index. This study defined case retrieval as an action for conducting time-series analysis on the semantic meanings of the Chinese time period (also known as the Chinese sexagenary cycle) and then choosing the cases that meet the semantic time period and location as the candidate cases. This is also the biggest difference between this study and traditional information systems. With the support of semantic technology, the operating interface of this system automatically offers self-explanatory functions. For example, in Table 1, the contents of Start_DT (2010/10/1 12:00:00 AM) and End_DT (2010/10/1 02:00:00 AM) were converted into the semantic contents shown in Table 2 (DT_Seasons, DT_Months, DT_Days, DT_Weeks, DT_Sesions, and DT_Dizhi) based on the Chinese sexagenary cycle.

Table 1. Example-data content.

Start DT	End DT	Area Type Name	Area Size Name	MAC Tpye Name	Sensor Data
2010/10/1 12:00:00 AM	2010/10/1 02:00:00 AM	Computer Classroom	Large	Humidity, Temperature, Illumination, CO ₂	54.000, 18.000, 463.000, 921.000

Table 2. Example-semantic meaning at the time period.

DT Seasons	DT Months	DT Days	DT Weeks	DT Sesiions	DT Dizhi
Fall	October	Early Month	Friday	AM	First

Stage 2 was the process to retrieve the similar case. It was divided into two steps: (1) Directly compare the values of case; if the case was the same, then the similarity calculation was skipped and this candidate case was deemed a solution; and (2) if the case was not the same, then it entered into similarity calculation. This study adopted the most common Nearest-Neighbor Retrieval method for similarity calculation because this method was simple. As long as the similarity between cases can be analyzed, the purpose of retrieval can be achieved. The calculation of Equation (1) is:

$$SR(P, C) = \sum_{i=1}^n [f(P_i, C_i)], \tag{1}$$

where P denotes the problem; C is the original case in the database; n is the number of attributes for each case; i is the order of the attribute (from 1 to n); $f(P_i, C_i)$ is the problem and original case in the database belonging to the i th attribute; and $SR(P, C)$ means the similarity between the problem and original case in the database (Similarity, SR).

The Nearest-Neighbor Retrieval is quite simple and intuitive. The system compared the characteristic attributes of a new problem and the original cases in the case base, obtaining a complete case similarity value. The calculated similarity value was used as a basis for the priority of case usage. The equations for related distance conversion, normalized distance, and similarity are shown respectively as Equations (2)–(4).

$$d_i(P, C) = \sum_{i=1}^n \left| \frac{V(P_i) - V(C_i)}{V(C_i)} \right|, \tag{2}$$

The distance between the problems and the characteristic attribute of the cases in the case base is called $d_i(P, C)$. Here, P denotes the problem; C is a specific case in the recommended solution; $V(P_i)$ is the value of characteristic attributes of the problems; and $V(C_i)$ is the value of characteristic attributes of the original cases in the case base. Equation (2) is an extremely traditional method of calculating distance functions. In order to improve the accuracy of case similarity, the weight (W) of each characteristic attribute was added to make the judgment. Through the calculation of the weight and normalized distance of the problems and the original cases in the case base, a more accurate judgment can be obtained. However, in order to take operation effectiveness and reaction time of the intelligent energy-saving information system into account, this system did not use the weight as a calculation factor. The calculation equation is shown as:

$$d_{sum}(P, C) = \frac{\sum_{i=1}^m d_i(P, C)}{m}, \tag{3}$$

where $d_{sum}(P, C)$ is the normalized distance between the problems (P) and cases (C). The normalized equation adopted by the system is dividing the distance between each characteristic attribute by m characteristic attributes.

The system can finally get the difference between cases by the normalized distance; that is, a longer distance represents greater differences between cases. Inferring this meaning by logic, it means that a shorter distance implies a more similar case. Therefore, the similarity value of the case can

be obtained by making a contrast of the distance. The similarity value is equal to one minus the normalized distance and then multiplied by 100%, as shown by:

$$SR(P, C) = (1 - d_{sum}(P, C)) \times 100\%, \tag{4}$$

When the identical case or superset was found in “case retrieval”, then “case reuse” directly became the recommended solution for the case. If there was no identical case in case retrieval and only similar cases could be given to case use, then these similar cases could not be a solution due to certain conditions or relation limitations. However, these similar cases may be referable and may be the final solution if an appropriate adaptation was given, as shown in Table 3.

Table 3. Case similarity.

Similarity	Representative Value (SR(V))	Recommended Case
High (zero adaptation)	$0.85 \geq SR(V) > 1$	V
Average (substitutional adaptation)	$0.75 \geq SR(V) > 0.85$	V
Low (seek external experts)	$SR(V) < 0.75$	X

Note: SR(V) represents similarity value, and 0.75 is the threshold of similarity.

When the system entered the process of case reuse, it may encounter five situations, as shown in Table 4. The situation “single-much like” was the recommended solution; for the case of “multiple-much like”, the system chose the one with the highest similarity as the recommended solution; for a “single-like” case, it entered the procedure “case adaptation”, and the adapted case was used as the recommended solution. It is highly possible that a case fails to meet the similarity threshold. For such a situation, the system would seek assistance from external experts. In the case of “single or multiple-like”-that is, its similarity value was between 0.75 and 0.85 ($0.75 \geq SR(V) > 0.85$) (0.75 is the system condition threshold)-the candidate case must be appropriately adapted. This process is an important part of the case-based reasoning, which is called “case adaptation.”

Table 4. Reuse application-types of problems.

Types of Reuse	Whether to Take It as a Solution	Whether to Adapt It	Solution	Whether to Take It as a Solution
single-identical ($SR(V) = 1$)	Yes	-	-	-
multiple-identical ($SR(V) = 1$)	Yes (Randomly select 1 by the system)	-	-	-
single-much like ($0.85 \geq SR(V) > 1$)	Yes	-	-	-
multiple-much like ($0.85 \geq SR(V) > 1$)	Yes (System selects the case with the highest similarity value)	-	-	-
single-like ($0.75 \geq SR(V) > 0.85$)	No	Yes	Transformational adaptation	Yes
multiple-like ($0.75 \geq SR(V) > 0.85$)	No	Yes	1. Transformational adaptation 2. Select the case with the highest similarity value as the recommended solution	Yes
Does not meet the threshold ($SR(V) < 0.75$)	No	No	Experts seeking	Yes

Note: SR(V) represents similarity value; “single-” means there was only one candidate case; “multiple-” represents there were several candidate cases; “-” means no action.

The degree of “case adaptation”, also known as “case revision”, is between case reuse and case trimming, and this study categorized the case through the application of case revision. In past studies, scholars have summarized a variety of adaptation methods. Considering the applicability of the field related to this study and other considerations, this study chose the substitutional adaptation as the method. There were three main reasons for choosing this method.

- First, the advantage of the substitutional adaptation method is that it is not necessary to make any changes to the original case structure. For a case-based system, this method can easily solve similar new problems in the case architecture. Hence, such an adaptation method is most efficient. Taking an energy-saving system as an example, this case-based reasoning mechanism, with the characteristic of cycling, was used in a real-time environment, and using substitutional adaptation could reduce the burden on the system operation.
- The similarity value of the case retrieval in this study was calculated by a single type of sensor, and the average value of the sum was the similarity value of the case. Therefore, in the application case adaptation, a single type of sensor and its corresponding values were used as the basis of the index, and the values were slightly adjusted in an appropriate manner. Until the end, the case with the highest similarity value was taken as the recommended case. This method did not perform actions that would damage the case structure, such as adding or deleting columns. Instead, it just made a slight adaptation to the values of the sensors.
- This study used the temporary table provided by .Net strategy to improve the problems in RAM. The action was to take SQL (Structured Query Language) to structure the query language through .Net. SQL is a programming language for querying, updating, and managing relational databases. SQL SELECT statement allows specific data to be easily retrieved, sorted, and filtered. Because of these three reasons, this study chose the substitutional adaptation method.

Table 5 is the case adaptation, which is an analysis of a single type of sensor. First, the necessary condition for adaptation was that the overall similarity value of the problems and the cases of database shall be $0.75 \geq SR(V) > 0.85$. Moreover, this study managed three types of adaptation: adaptation type 1, adaptation type 2, and adaptation type 3. The purpose of adaptation of this study was to improve the similarity between the new problems and the cases so that the new problems were used as the basis of the slight-adapting method.

Table 5. Case adaptation-analysis of a single type of sensor.

Adaptation Condition	Adaptation Type 1	Adaptation Type 2	Adaptation Type 3
single (multiple)-like ($0.75 \geq SR(V) > 0.85$)	$V_C > V_P$	$V_C = V_P$	$V_C < V_P$

Note: $SR(V)$ is the similarity value of new problems (P) and cases (C); V_P is the value of new problems; V_C is the value of cases.

Figure 10 is a demonstration of slight adaptation in which a certain type of the value of cases (V_C) was larger than the value of problems (V_P). Its challenge was how to shorten the distance between these two values. In this study, the average value was used to adapt V_C . The new value generated after adaptation was named V_{AV} , as shown in Equation (5). Equation (6) is the average of characteristic attributes (i) of the problems and the cases, which also represent the solution of the new adapted cases. In this equation, P represents the problems; C means a specific case in the recommended solution; $V(P_i)$ represents the value of characteristic attributes of the problems; and $V(C_i)$ means the value of characteristic attributes of the cases. Equation (6) is the average conversion formula that could get the solution content of new case C_{Ad} after adaption.

$$V_{AV} = (V_P + V_C)/2, \tag{5}$$

$$C_{Ad} = AV_i(P, C) = \sum_{i=1}^n \left(\frac{V(P_i) + V(C_i)}{2} \right), \tag{6}$$

where C_{Ad} is the new case that has been adapted; and $V_i(P,C)$ is the problems and cases of the i th attribute.

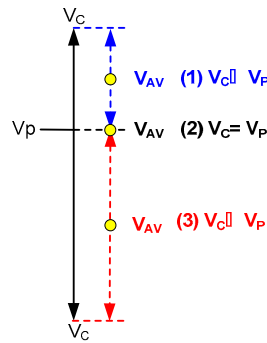


Figure 10. Case adaptation-analysis of a single type of sensor.

By substituting C_{Ad} into Equations (2)–(4), the new similarity values after adaptation can be calculated, and the one with the highest new similarity value was the recommended solution for energy-saving operation. The case adaptation of this study (substitutional adaptation) was a slight adaptation of values based on not changing the case structure. In this adaptation, the new problems (each type of sensors and their corresponding values) and the candidate cases (the corresponding type of sensors and values) were adapted slightly, so as to improve the similarity between cases.

This study defined the application “case retainer” as the way to generate independent-periodical cases to explore a distinctive method of retaining cases, i.e., this is one of the important steps in sustainable processing. The huge historical data in the raw database were fully used, and so this study transformed the case retainer from a real-time application into a periodical independent application, which was different from the characteristic of cycling of the traditional case-based reasoning method. In short, to realize the difference between this case-based reasoning system and traditional case-based reasoning systems, we create an application that is particularly suitable for the energy-saving information system of this study. Figure 11 shows the operational flow of the case retainer. In this flow, the case monitor periodically retrieves the “unconverted” historical information (raw data) to WIAS with a system-defined cycle time (like two hours). The time was then converted into semantic words based on the Chinese sexagenary cycle, and via sorting, the common information was taken out as useful information. Finally, the relevant information was converted into semantic words, denoting the action in which the semantic case was retained into the case base has been completed.

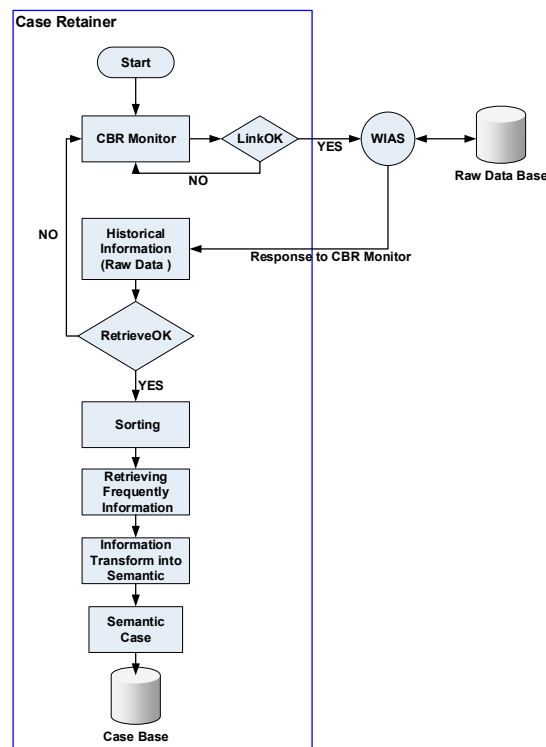


Figure 11. Operational flow of the case retainer.

4. System Presentation and Effectiveness Analysis

4.1. System Presentation

The development environment and tools of the proposed system include Java SE Development Kit, AppServ, JCreator LE, Microsoft Visual Studio, Microsoft SQL Server, etc. The back-end servo system is IBM System x3500 M2.

At the beginning of system operation (periodic off-line), the researcher selected the case monitor page, entered the time (months, days, or hours) of the conversion in the case base, and carried out the detection. When the detection was completed, the total number of times of the processing by the case monitor was displayed by the “Count” in the screen, and each step in the process of off-line operation was presented, including making an “inquiry” on environmental sensing data that has not been converted into a case (Figure 12). Next, the data were “converted” via defining the contents with semantic meaning (Figure 13). In the end, the Sensor_Data and its corresponding operation mode were retrieved, and the cases were integrated and stored into a “case base” in accordance with the types of the sensor, such as Humidity, Temperature, Illumination, and CO₂ (Figure 14).

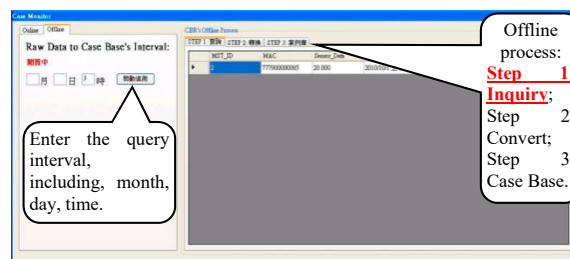


Figure 12. Off-line: Inquiry.

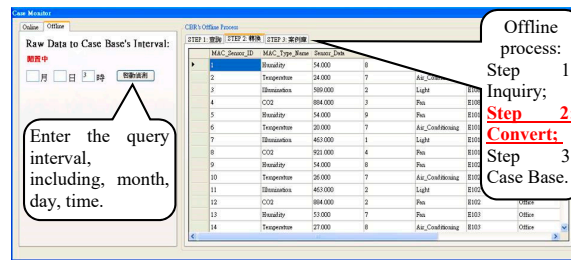


Figure 13. Off-line: Convert.

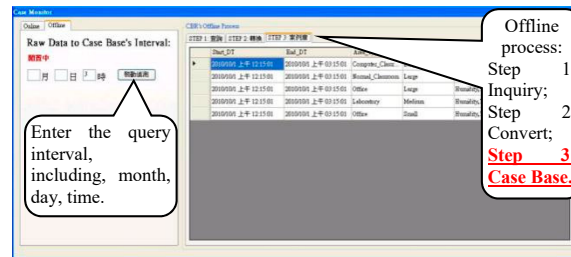


Figure 14. Off-line: Case Base.

As described in the aforementioned three-stage intelligent decisions, OntoDMA processed the query information that did not appear frequently or did not exist. At that time, OntoCBRA only launched the corresponding web services CBR_InsTmpCaseData provided by WIAS and recorded the information value of this sequence to Case_TMP to complete the processing. If OntoDMA was unable to carry out the processing, then OntoCBRA would be triggered by the solution finder of Ubi-IA. Through the web services CBR_Solutions offered by WIAS, OntoCBRA then launches the online case application. This system sorted all of the “MAC_Type_Name” in this space in accordance with Area_Name, Area_Type_Name, and Area_Size_Name and calculated the average values of “avgVALUE”, “avgvRangeUp”, and “avgvRangeDown”. In order to expand the range of case retrieval, this system used “avgVALUE” as the base. Multiplying the value of “avgVALUE” by 75% became “avgvRangeDown”, while multiplying by 125% became “avgvRangeUp”. The design of avgvRangeDown and avgvRangeUp can avoid the situation in which no case is available when using avgVALUE as the base. In other words, the range of case retrieval equals the range between avgvRangeDown and avgvRangeUp (Figure 15).

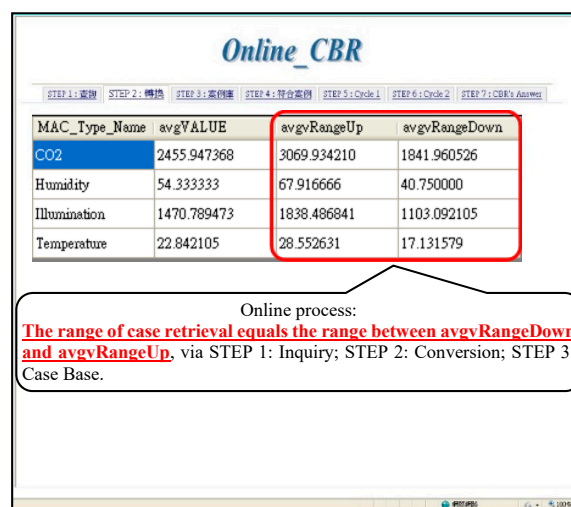


Figure 15. Case retrieval range.

In the sector of case retrieval, OntoCBRA took avgvRangeUp and avgvRangeDown as the threshold of similarity (75% to 125%). Figure 16 shows all cases that met the threshold. If no case met the similarity threshold, then it means that this system had no experience on processing a similar situation, and CBR Agent sent a null response to Ubi-IA. With the support of the system ontology database, this system disassembled the cases meeting the threshold and computed the similarity between the data of these cases and the sequence monitoring data in Case_TMP, as shown in Figure 17. If the data of these two were “identical”—that is, the value of similarity was 1 (100%)—then this system would directly return the answer of this case to Ubi-IA as the solution of the sequence information value. The step of case reuse was thus completed if the two were “partially similar but not identical”, then this system entered into the sector of case trimming.

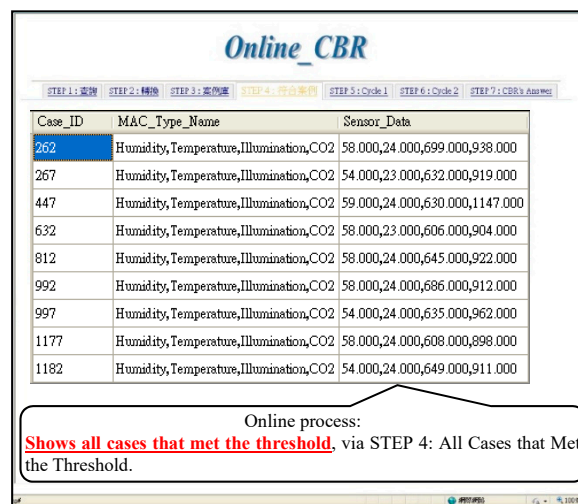


Figure 16. All cases that met the threshold.

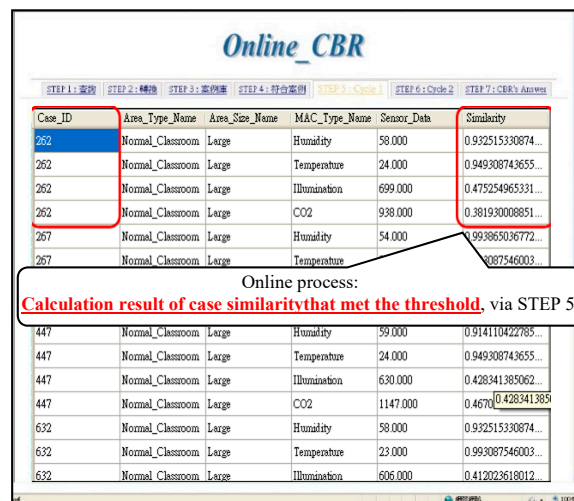


Figure 17. Calculation result of case similarity.

If there was no identical case in the step of OntoCBRA, then the system would trim the complete similarity of cases, as shown in Figure 18. The column “SimilarityCount” indicates the sum of the “Similarity” of the same “Case_ID”, while the column “SimilarityAvg” represents the average value of the similarity. At the end of the step of case trimming, the system selected the Case_ID with the highest similarity as the output of OntoCBRA in the trimming step (Figure 19) and returned this output to Ubi-IA as the solution of the sequence information value. In terms of case retainer, if the case difference,

which is the reverse of the similarity of a trimmed case, hit the system threshold, then the case would be stored in the case base. The difference was set externally by domain experts.

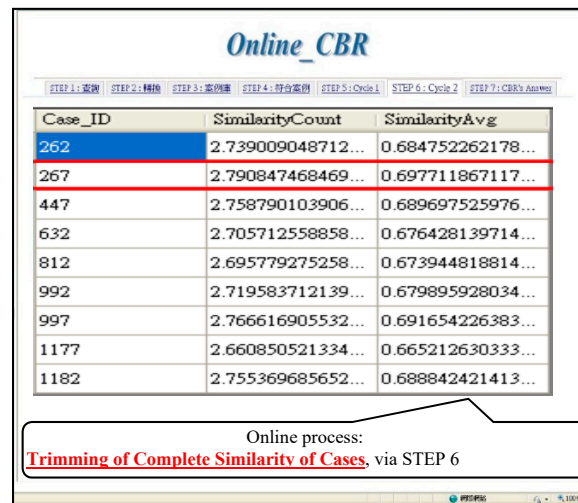


Figure 18. Trimming of complete similarity of cases.

57	2016/2/16 上午12:00:00	2016/3/1 上午1...	Office	Large	Humidity, Tempe...	43,28,1078,13
58	2016/2/16 上午12:00:00	2016/3/1 上午1...	Office	Small	Humidity, Tempe...	40,16,1311,1541
59	2016/2/16 上午12:00:00	2016/3/1 上午1...	Laboratory	Medium	Humidity, Tempe...	39,2,1117,2568
60	2016/2/16 上午12:00:00	2016/3/1 上午1...	Laboratory	Small	Humidity, Tempe...	46,10,1814,2598
61	2016/10/1 上午12:00:00	2016/10/16 上...	Computer_Class...	Large	Humidity, Tempe...	52,30,950,1680

Figure 19. Output in the trimming step.

4.2. System Effectiveness Analysis

The experimental content was utilized to test the performance of OntoCBRA and its learning effectiveness. However, this study used the similarity value threshold as the trigger for case reuse, case trimming, and experts seeking. It was thus necessary to conduct the analysis with the usage rate of different similarity values at each level, carry out the on-board testing over a long term, and adjust the threshold of similarity value. At the end of the analysis, the similarity value threshold was summarized, as shown in Table 6. This threshold was applied to the complete energy-saving information agent system with the front-end and the back-end to analyze the performance of the system [20].

Table 6. Similarity value threshold.

	Level 1 Identical	Level 2 Case Reuse	Level 3 Case Trimming	Level 4 Experts Seeking
Scope of testing	100%	100% > SR(V) ≥ 85%	85% > SR(V) ≥ 75%	SR(V) > 75%

First, environmental sensing data from the energy-saving demonstration classroom in the Electrical Engineering and Computer Science Building of St. John’s University were collected in 2011. In total, 68,000 or more monitoring records were collected and converted into the internal record format and stored in the system case database, shown as Figure 20. The first experiment focused on the performance effectiveness of OntoCBRA and used the aforementioned records as its training dataset. Based on the “CBR-TransRawtoCase” service provided by the WIAS web service, the system established a total of about 4000 cases in the case base. From these test data, 6500 queries were randomly selected as samples to test the performance of OntoCBRA. The results of the five experiments were examined by domain experts, who are energy saving experts, to determine whether the reasoning results of the proposed system meet the definition and requirements of energy saving. As shown in Table 7, this system had four levels: (1) Direct use of the recommended cases (with 100% similarity); (2) case

reuse (with similarity over 85% but less than 100%); (3) case trimming (with similarity over 75% but less than 85%); (4) experts seeking (with similarity less than 75%). Table 7 shows that for an average of 2520.4 queries, Level 1 can process 50.2 queries on average, while the average queries processed by Level 2, Level 3, and Level 4 are 1310.4, 914.6, and 243.4. The result proved that OntoCBRA processes an average of about 40% of the inquiries, leaving only about 60% of the inquiries to the back-end system for processing, thus effectively reducing the common burdens of back-end server systems.

Figure 20. Original system case database.

Table 7. Analysis results of performance effectiveness.

Test Order	Total Number of Inquiries	Level 1 Identical	Level 2 Case Reuse	Level 3 Case Trimming	Level 4 Experts Seeking	Total Number of the Inquiries of OnotCBRA	The Amount That Has Completed Processing in the System %
1	6301	51	1288	913	270	2522	40.0
2	6227	67	1339	901	210	2517	40.4
3	6111	49	1298	927	243	2517	41.2
4	6289	35	1339	897	249	2520	40.1
5	6157	49	1297	935	245	2526	41.0
average	6217	50.2	1310.4	914.6	243.4	2520.4	40.5

The second experiment focused on the learning effectiveness of OntoCBRA cases. This study used the same system as the first experiment and the cases in the case reuse experiment 1 and subsequently collected more than 34,000 monitoring records from the 2016 records. This study then randomly selected 6500 records from these materials to conduct five environmental query experiments. Table 8 shows the results of the five experiments that were reviewed by domain experts one by one. In Table 8, for an average of 2534.4 inquiries, 53.8 inquiries were completed at Level 1, 1342.6 inquiries at Level 2, 901.8 inquiries at Level 3, and 236.2 inquiries at Level 4. An average of 41% of inquiries was completed in OntoCBRA.

In summary, in the two experiments of OntoCBRA, the effectiveness at Level 1 improved by 7.2% ((53.8 – 50.2)/50.2 × 100% = 7.2%), the number of inquiries at Level 2 increased by 2.5% ((1342.6 – 1310.4)/1310.4 × 100% = 2.5%), the number of data trimming at level was reduced 1.4% ((901.8 – 914.6)/914.6 × 100% = –1.4%), and the external query also decreased 3.0% ((236.2 – 243.4)/243.4 × 100% = –3.0%). Although the effectiveness was slightly improved, the improvements in effectiveness of Level 1 and Level 2 meant the system accumulated more experience in processing. In addition, the decreases of case trimming and external query at Level 3 and Level 4 proved the learning effectiveness of this system. This study believes that the ability of the system’s case solving will get better over time. As the content of the case base becomes more comprehensive, the ability to solve problems of this system will increase. If the burdens at Level 3 and Level 4 can be reduced, then the

effectiveness of OntoCBRA will be better, and the operational burden of the back-end database could be diminished simultaneously.

Table 8. Analysis results of learning effectiveness.

Test Order	Total Number of Inquiries	Level 1 Identical	Level 2 Case Reuse	Level 3 Case Trimming	Level 4 Experts Seeking	Total Number of the Inquiries of OnotCBRA	The Amount That Has Completed Processing in the System %
1	6144	60	1297	903	259	2519	41.0
2	6330	71	1345	879	237	2532	40.0
3	6007	50	1301	923	249	2523	42.0
4	6394	37	1411	890	207	2545	40.0
5	5997	51	1359	914	229	2553	42.6
average	6904.6	53.8	1342.6	901.8	236.2	2534.4	41.1

5. Discussion and Conclusions

This study developed a cloud energy-saving case-based reasoning agent built on the support of web services, ontology, and big data analytics for sustainability science. It showed the relevant R&D technologies and the results in great detail to students for educational purposes. It reviewed the relevant technologies for building a web services platform and explored how to widely integrate and support the cloud interaction of the back-end agent system via the technologies. After reviewing the R&D screen and the experiments of this system, the findings show that OntoCBRA can process over 40% of the inquiries, leaving only under 60% of the inquiries to the back-end system for processing. OntoCBRA not only can accumulate the experiences in data processing but can also effectively reduce the common burden of back-end server systems. In addition, the performance at querying new problems was improved. Importantly, according to the results of the 2013 paper [15], the main system architecture and its four main steps are unchanged, but two optimization steps for calculating the similarity of the cases are added to more clearly illustrate the relevant calculation procedures. This not only optimizes the overall efficiency but also allows the reader to repeatedly implement the detailed process of the proposed system, which are the contributions of this paper. Finally, we also applied the 2013 cases library directly to the 2016 data, which not only presented the four major steps of cases inference processing verification but also achieved about 5% system processing performance improvement. So far, although this study has developed an OntoCBRA system that works properly, there are still many parts that are not completed and need to be improved, as explained below.

- (1) The trigger for case reuse, case trimming, and expert-seeking in this study is currently based on seeking external experts. Future studies can explore the feasibility of automatically adjusting the trigger.
- (2) The independent case retainer meant that accumulating cases (case generation) is at the core of case-based reasoning. Since the cases in the current case base are quite few, this study only designed the function of adding cases. However, if future researchers only add cases to increase the knowledge contents, then the case base will have too many unnecessary similar cases, resulting in an increase in system burden. Hence, how to reduce the burden of the case base may be an issue worth exploring in the future.
- (3) Case adaptation is presently based on the system-defined similarity threshold. In the future, researchers can explore the feasibility of deeming the correlation and toughness between the type of sensor, electrical equipment, power consumption, equipment operation, and sensing data as the adaptation conditions.
- (4) This study concluded after comparing the new problem and the candidate cases that the case with the highest similarity was the recommended case. However, if there is no identical case after the comparison between a new problem and candidate cases, is it possible to consider the

satisfaction of the candidate case as the judgment for the recommended case? It is suggested that future studies can carry out different recommendation methods.

In practice, this study built a cloud energy-saving case-based reasoning agent that integrates multiple intelligent technologies. The development of this agent system relied on the support of web services, ontology, and big data analytics under a cloud environment. This study not only effectively implemented the practical application of OntoIAS but also successfully applied the case-based reasoning technique to the learning performance of the system. Hence, this study has a certain degree of software economic benefits and broad application prospects and can effectively continue with the R&D results of related projects. Academically, this study developed an intelligent cloud energy-saving data processing and decision-making agent. The capability, architecture, and technology involved in this system demonstrate the sound application and integration of case-based reasoning and related technologies as applied to smart environments. This study is unique and innovative and should help contribute to the development of related industries and countries. The authors plan to explore further research from the perspective of vocational colleges and universities, so as to make a contribution toward practical applications and technology integration.

Author Contributions: This research article contains two authors, including S.-C.C. and S.-Y.Y., S.-C.C. and S.-Y.Y. jointly designed the overall architecture and related algorithms, and also conceived and designed the experiments; however, S.-Y.Y. coordinated the overall plan and direction of the experiments and related skills; S.-C.C. and S.-Y.Y. not only contributed analysis tools but also analyzed the data; S.-C.C. performed the experiments; and S.-Y.Y. wrote this paper and related reply. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Ming-Yu Tsai and Guo-Jui Wu for their assistance in earlier system implementation and preliminary experiments. This research is sponsored under grants 106-2221-E-129-008 and 107-2632-E-129-001 by the Ministry of Science and Technology, Taiwan, R.O.C. The authors feel deeply indebted to the Department of Fashion Administration and Management and Department of Information and Communication Engineering, St. John's University, Taiwan, for all aspects of assistance provided.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, S.F. Utilizing Case-Based Reasoning Method in Parameter Planning for Fused Deposition Modeling Process of 3D Printing. Master's Thesis, Department of Mechanical Engineering, National Pingtung University of Science and Technology, Pingtung, Taiwan, 2018.
2. Lin, S.X. Design Thinking for Developing a Case-Based Reasoning Emotional Robot: In the Scenario of Interactive Interview. Master's Thesis, Department of Interaction Design, National Taipei University of Technology, Taipei, Taiwan, 2017.
3. Ministry of Economic Affairs, Taiwan. 2017 Year of Energy Supply and Demand Overview. Available online: https://www.moeaboe.gov.tw/ecw/populace/content/ContentDesc.aspx?menu_id=865 (accessed on 22 July 2019).
4. Vafeiadis, T.; Nizamis, A.; Apostolou, K.; Charisi, V.; Metaxa, I.N.; Mastos, T.; Tzovaras, D. Intelligent Information Management System for Decision Support: Application in a Lift Manufacturer's Shop Floor. In Proceedings of the 2019 IEEE International Symposium on Innovations in Intelligent Systems and Applications, Sofia, Bulgaria, 3–5 July 2019; pp. 1–6.
5. Chen, S.W. Risk and Outcome Analysis in Patients with Cardiac Surgery, Effect of Lumican in Cardiovascular Disease: A Project from a Big Database to Molecular Cardiology. Ph.D. Thesis, Graduate Institute of Clinical Medicine Sciences, Chang Gung University, Taoyuan, Taiwan, 2018.
6. Lin, H.Y.; Yang, S.Y. A Cloud-based Energy Data Mining Information Agent System Based on Big Data Analysis Technology. *Microelectron. Reliab.* **2019**, *97*, 66–78. [[CrossRef](#)]
7. Yu, C.J. A Recommendation Information System for Selecting the Most Efficient Milling Machine by Using Case-Based Reasoning Technology of Machine Learning. Master's Thesis, Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan, Taiwan, 2018.

8. Lou, Y.P. A Study of Credit Default Risk Evaluation Model Construction Based on Case-based Reasoning. Master's Thesis, Department of Information Management, National Defense University, Taoyuan, Taiwan, 2017.
9. Huang, H.J. Applying Case-Based Reasoning to Forecast the Sales of Sport Shoes. Master's Thesis, Department of Distribution Management, Takming University of Science and Technology, Taipei, Taiwan, 2016.
10. Relich, M.; Pawlewski, P. A case-based reasoning approach to cost estimation of new product development. *Neurocomputing* **2018**, *272*, 40–45. [[CrossRef](#)]
11. Faia, R.; Pinto, T.; Abrishambaf, O.; Fernandes, F.; Vale, Z.; Corchado, J.M. Case based reasoning with expert system and swarm intelligence to determine energy reduction in buildings energy management. *Energy Build.* **2017**, *155*, 269–281. [[CrossRef](#)]
12. Aljuboori, A.S.; Meziane, F.; Parsons, D.J. A new strategy for case-based reasoning retrieval using classification based on association. In Proceedings of the 12th International Conference on Machine Learning and Data Mining in Pattern Recognition, New York, NY, USA, 16–21 July 2016; pp. 326–340.
13. Chung, I.L.; Chou, C.M.; Hsu, C.P.; Li, D.K. A programming learning diagnostic system using case-based reasoning method. In Proceedings of the 2016 International Conference on System Science and Engineering, Nantou, Taiwan, 7–9 July 2016; pp. 1–4.
14. Yang, S.Y. A Novel Cloud Information Agent System with Web Service Techniques: Example of an Energy-saving Multi-agent system. *Expert Syst. Appl.* **2013**, *40*, 1758–1785. [[CrossRef](#)]
15. Yang, S.Y. Developing an Energy-saving and Case-Based Reasoning Information Agent with Web Service and Ontology Techniques. *Expert Syst. Appl.* **2013**, *40*, 3351–3369. [[CrossRef](#)]
16. Chen, K.Y.; Chen, F.H.; Yang, S.Y. An Upgraded Intelligent Mobile Information Multi-Agent System with Universal Application Interfaces based on Open Data of Taiwan Government. *J. Comput. Appl. Sci. Educ.* **2017**, *4*, 1–23.
17. Chen, W.C.; Chen, W.H.; Yang, S.Y. A Big Data and Time Series Analysis Technology-based Multi-Agent System for Smart Tourism. *Appl. Sci. Open Access J.* **2018**, *8*, 947. [[CrossRef](#)]
18. Chiu, J.C. The Application on Big Data Analysis of R and Hadoop. Master's Thesis, Department of Statistics and Information Science, Fu Jen Catholic University, New Taipei City, Taiwan, 2014.
19. Yang, S.Y. OntoIAS: An Ontology-Supported Information Agent Shell for Ubiquitous Services. *Expert Syst. Appl.* **2011**, *38*, 7803–7816. [[CrossRef](#)]
20. Tsai, M.Y. Developing a Case-Based Reasoning Agent System with Web Service Techniques for Energy-Saving Information. Master's Thesis, Department of Computer and Communication Engineering, St. John's University, New Taipei City, Taiwan, 2013.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).