IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON COOPERATIVE AND INTELLIGENT SENSING

Received January 20, 2017, accepted February 5, 2017, date of publication February 8, 2017, date of current version March 15, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2666419

Context-Aware Scheduling in Personal Data Collection From Multiple Wearable Devices

AO GUO, (Student Member, IEEE), and JIANHUA MA, (Member, IEEE)
Graduate School of Computer and Information Sciences, Hosei University, Tokyo 184-8584, Japan
Corresponding author: A. Guo (guo.ao.33@stu.hosei.ac.jp)

ABSTRACT Due to the prevalence of smartphones and various wearable devices, the collection of rich personal data that can be used for human activity recognition, user modeling, and personalized services has become feasible. Because of its popularity and high accessibility, the smartphone has not only become an effective terminal in personal data collection, but also a gateway connecting wearable devices and gathering various types of personal data from these wearables. In most current applications, such wearables operate to collect data according to a fixed schedule, often preset manually by a user. The main problems in the data collection arising from such fixed scheduling are weak adaptiveness to wearables' state changes, a high level of redundancy in collected data, and possible mismatches in the dynamic precision requirements of data capture. Therefore, we propose a context-aware scheduler, that is able to dynamically adjust a data collection schedule based on contingent situations in the condition of wearables, system resource availability, and user behavior. This paper is focused on context data detection and data collection scheduling in a smartphone-based client-server system. The smartphone functions as not only a gateway gathering data from multiple wearables, but also a terminal for the performance of a context-aware scheduler. A context-aware engine is implemented to handle different contextual information. The data quality and system performance have been evaluated and verified in practical experimental tests.

INDEX TERMS Context-aware, smartphone, data collection, schedule, wearable, adaptiveness, user pattern.

I. INTRODUCTION

Due to the rapid progress of information and communication technologies (ICT), various digital explosions in terms of data, connectivity, service and intelligence explosion are emerging in an integration of the cyber and physical world, the hyper world [1], [2]. Ubiquitous/pervasive computing, Internet of things (IoT), cyber-physical system (CPS) and others can be seen as means to practically bridge and even merge the digital cyber world and the physical world for the formation of the hyper world [3], [4]. In this new world, numerous items, including various devices, machines and persons are interconnected, and these are able to compute and also generate huge amounts of data, i.e., big data, day by day. Based on our study of Cyber-I [5], a novel wearable system called Wear-I (Wearable Individual) has been proposed recently as a new paradigm for wearable computing in personal data collection and personalized service provision [6]. In a Wear-I, multiple wearable devices are used to collect personal data from sensors embedded in these wearables.

As an increasing number of wearable devices appear in our ambient environment, a great deal of personal data could be collected by a smartphone, making further personality computing possible [7]. As the smartphone comes to play a greater and greater role in human life, such a smartphone-based and wearable-combined personal data collection system could exhibit two basic features, dynamic resource utilization and variable personal data requirement, in terms of different application scenarios. Two representative scenarios are shown in Fig. 1. A user checks their email at home with wearing a bracelet and an LG watch in the first scenario. While in the other scenario, a user takes a walk to an office wearing an LG watch and a ring. It can be seen that both the wearable devices and smartphone resources change accordingly in the two scenarios.

Concerning the features mentioned above, two fundamental issues are (1) how to collect and manage data from various wearables effectively under different scenarios, and (2) how to use the limited system resources in a smartphone efficiently. In our previous research, a smartphone-based system was developed with fixed scheduling for personal data collection [8]. Such a fixed schedule preset by a user adapted poorly to different scenarios and proved to be inflexible. Therefore, this paper presents a context-aware scheduler that is able to dynamically adjust a data collection schedule based

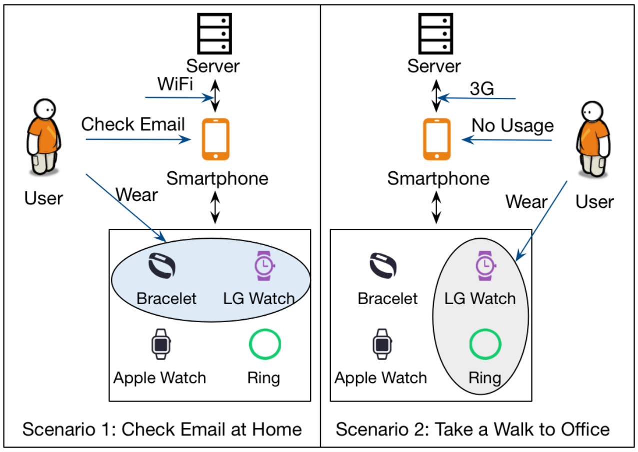

FIGURE 1. System conditions in two different scenarios.

on contingent situations in the condition of wearables, system resource availability and user behavior.

Hence, the first research objective in this study was to rebuild the system's architecture and functions specifically to adapt to wearables' changing states and their heterogeneous data types. The data collector aims to collect and store data, while the data preprocessor is in charge of fundamental processes i.e. data type normalization and data redundancy elimination. The data uploader is to send processed data to a server.

Our next research objective was to convert the fixed scheduler into a context-aware scheduler to collect data according to various contexts. To fully evaluate possible factors that may impinge on data collection, three contexts were used in this study; the wearable context, the system context and the user context. These sensed the state of the wearable, system resource utilization, and user behavior patterns, respectively. Context detection is one of the key techniques necessary for measuring each of these contexts.

A third research objective was to achieve high wearable data quality and good system performance [9], [10]. Data Quality (DQ) is judged according to three criteria; namely, data accuracy, (DA) referring to the degree to which a user's state is described, Data Integrity (DI), referring to the number of possible data types that could be collected by certain wearables, and data efficiency (DE), referring to the non-redundancy of collected data. System performance (SP) is assessed using two criteria, namely power consumption and resource utilization. To realize optimal data quality and system performance, a context-aware engine embedded in the context-aware scheduler is implemented according to three different models, namely a system power consumption model, a multiple resource utilization model and a data quality optimization model.

The remainder of this paper is organized as follows. The next section is about related studies and their relation to our research. Section III describes our system architecture and gives a general illustration of the proposed context-aware scheduler. Section IV explains how a context broker manages a context from assessing the three contexts mentioned above. Section V shows working flows in the context-aware scheduling process. Section VI explains the models in the context-aware scheduler in detail. Experimental results and their analyses are presented in Section VII. Conclusions are drawn and future work suggested in the last section.

II. RELATED WORK

Over a relatively short period of time, computing has come to focus on the personal in human life. Human-centered computing (HCC) aims to bridge the gaps between various disciplines such as signal processing, machine learning and ubiquitous computing involved with the design and implementation of the computing systems that support people's activities [11]. As a fundamental step for HCC, personal data collection and analysis is increasingly common nowadays following on from the prevalence of wearable devices [12]. As a result, a great deal of research is on, or related to, personal data collection. B. C. Singh, et al, developed a risk-benefit driven architecture to focus on protecting personal data through an approach that empowers individuals and measures possible risk and benefits during data release [13]. X. Zhou, et al, concentrate on personal data analytics to facilitate cyber individual modeling [14] by systematically organizing and refining a personal stream of data, which can help improve data processing and management in the CI-Spine tier and CI-Pivot tier of Cyber-I. E. Aïmeur and M. Lafond focus on internet personal data collection and voluntary information disclosure, with an emphasis on the problems and challenges facing privacy [15]. In contrast to these personal data related research efforts, which focus on personal data from the internet, our research is specific to the gathering of personal data from wearables under a data collection schedule. Such wearables could provide stable physical personal data continuously.

However, hardware limitations, in terms of battery life, CPUs, networks etc. restrict the possibility of unlimited data gathering. Therefore, by considering wearable battery levels and system resources utilization changing dynamically, a context-aware scheduler which senses various related factors would be ideal for personal data collection. Various categorizations of context have been proposed in the past. Schilit *et al.* [16] introduced the concept of context which is related to the location of nearby person hosts or objects, and how they change over time [17]. Hence, research related to the context above is regarded as research into context awareness. This is the first research into data collection with context awareness. C. Perera, et al, provided some context-aware computing methods for the Internet of Things by providing an in-depth analysis of context life cycle and evaluated a subset of projects which represent the majority of research and commercial solutions proposed in the field of context-aware computing conducted over the last decade [18]. Y. Lee [19] proposed a cooperative context monitoring system (CoMon+) for multi-device personal sensing environments. A benefit-aware negotiation method to

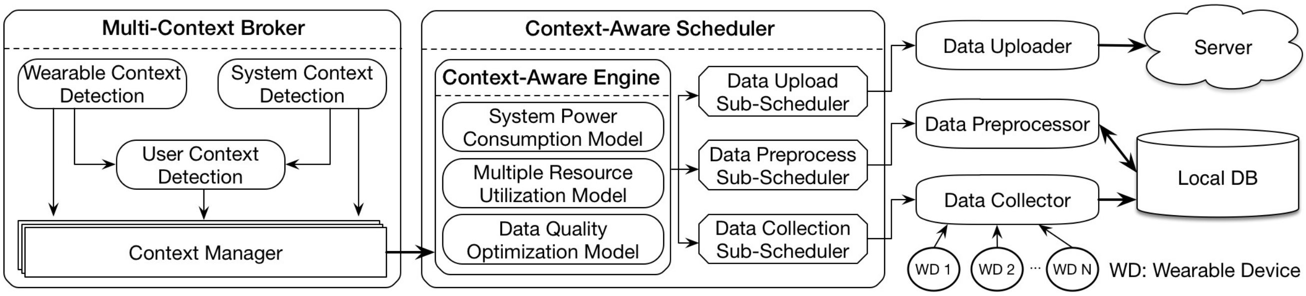

FIGURE 2. General process of context-aware data collection.

maximize the energy benefit of context sharing was developed during his research, and cooperators likely to remain in a vicinity for a long period of time were detected with CoMon+ [20]. Alternatively, T. Tran [21] focused on sensing interflow network coding and scheduling which adaptively encodes data across traffic to maximize a network's quality of service (QoS). Such a context-awareness concept is well suited for personal data collection under multiple wearable devices.

Similar to that research which focuses on data collection under different contexts, our research also schedules data collection under three contexts; a wearable context, system context, and user context. However, little research into context-awareness has considered users' influence on each context, i.e. location and event. The data collection system which was implemented in our previous research also ignored the problem of the influence of user behavior, which precludes simultaneous and consistent data quality and system performance. Therefore, we proposed a user context to describe a user's behavior pattern.

Some researchers are focusing on detecting and analysing the influence of users' behaviour. To analyse the exact influence, T. Mavridis, et al, provide a close-to-user propagation model to describe the users' influence in different scenarios [22], while Matsumura and Sasaki [23] focus on understanding leadership behaviour in human influence networks. Besides users' proactive behaviour, some inactive behaviour is also valuable to research. Dzvonyar, *et al.* [24] described a context-aware feedback system which consists of collecting user feedback enriched with usage context data, as well as a process for integrating such feedback into development activities. E. S. Marin and C. L. Carvalho [25] developed a number of models that maximize human influence of increasing the probability of finding reliable connections between numerous paths and interpersonal influences.

Different from the research little considering human behaviour influencing scenarios, we have further developed user behaviour patterns in this paper to classify user behaviour in three different aspects, namely user location pattern relating to spatial-temporal features [26], a user action pattern [27] and a battery charge pattern [28]. In order to achieve optimal data quality and system performance, system power consumption and multiple resource utilization are measured under these patterns respectively.

III. SYSTEM OVERVIEW OF CONTEXT-AWARE SCHEDULING

Our context-aware scheduling system has adopted the client/server mode, where a smartphone acts as a client embedded in a context-aware scheduler. The modules within the context-aware scheduling system are described below, and its architecture is illustrated in the second subsection.

A. GENERAL DESCRIPTION

As mentioned above, there are two distinct characteristics in the Wear-I system that may influence data quality and system performance. One is the use of multiple wearable devices that may be heterogeneous and subject to dynamic change. The other is the use of a mobile system that may only be able to carry out processes intermittently. Accordingly, the context detection in a multi-context broker is responsible for detecting these two situations. As shown on the left in Fig. 2, the wearable context detection and system context detection function inside the context broker is in charge of sensing the resource and basic state in each device. For example, the wearable wearing state and smartphone CPU utilization can be separately detected by these two context detections. However, some contexts which may have a great influence on personal data collection are hard to detect. Take the battery charging time as an example. The next battery charging event which could directly influence the battery life is hard to detect from these two contexts. Therefore, user context detection is set up to detect a user's behavior pattern constantly according to the wearable context and system context.

The context-aware scheduler is responsible for the potential factor analysis which may influence the process of data collection according to each context and then for further analysis using such factors in the resulting scheduling. The context-aware engine contains three kinds of models for independent contextual information; modelling in terms of system power consumption, multiple resource utilization, and data quality optimization. Meanwhile, these models could guide three sub-schedulers for personal data gathering, namely a data collection sub-scheduler, a data preprocess

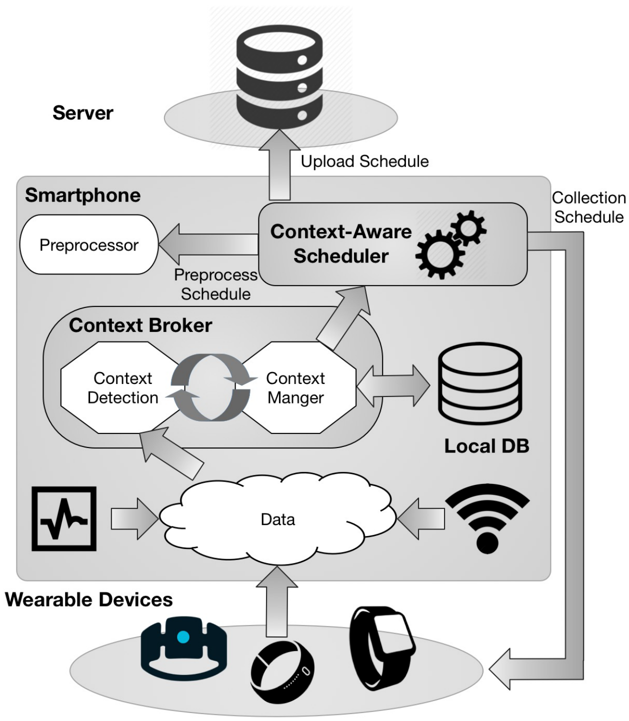

FIGURE 3. System architecture.

sub-scheduler and a data upload sub-scheduler. The detail of such a context-aware scheduler is introduced in section V.

B. SYSTEM ARCHITECTURE

For the purpose of clarifying the process in our smartphone-based system, the system architecture contains three parts, as shown in Fig. 3. For the first part, multiple wearable devices are grouped as the personal data source, as shown at the bottom of Fig. 3. The second part is the smartphone, which plays a role as a terminal to handle the schedule and preprocessing, as well as data upload, while the third part is a server, which receives the preprocessed data from the smartphone for permanent storage.

The major functions are implemented inside the smartphone as shown in the middle of Fig. 3. The total process is largely separated into three steps. Firstly, the data from the wearables and system resource information are gathered from the multiple wearable devices and system monitor respectively. Such data including the wearables' states, system resource utilization and the network state will then be sent to the context broker for the next process. Secondly, the context broker will start, and the detected context will then be transmitted to the context manager for storage and further provision of services. The multiple context information in terms of the wearable context, system context and user context is stored in a local database for further scheduling. As an engine of the whole data collection process, the context-aware scheduler conducts the final step for scheduling each process. The preprocessing, data collection and data upload is controlled accordingly by the context-aware scheduler. Consequently, the processed data is sent to and stored in a server database on the server side. After that, the process continues to the next round.

IV. MULTIPLE CONTEXT BROKER

As described in the last section, the multiple context broker primarily arranges the transactions between heterogeneous data and the context-aware scheduler to optimize multiple schedules. Three context detection methods for wearable, system and user are illustrated respectively in the following three subsections. The last subsection will introduce data management according to ontology classification.

A. WEARABLE CONTEXT DETECTION

As one of the fundamental context sources, a wearable device senses a lot of information which benefits the schedule adjustment. For example, the battery level of a certain wearable device can be measured to calculate the battery life. Consequently, the scheduler could adjust data collection frequency in order to provide a sustainable time of use.

The function of wearable context detection (WCD) aims to detect wearable data in three different aspects. The first is the wearable system condition, including the battery level, available sensor type, and connection status. WCD detects this system condition by directly mapping each datum with a fixed period. For instance, the wearable connection status can be easily detected by querying the Bluetooth pairing state with a certain system API. The second aspect is wearable sensor data. WCD serves to monitor each sensor data collection event and collecting state consistently. Basic information such as the wearable data throughput and each datum's time stamp could be recorded in detail, which could also be further detected by user context detection. The third is the activity state of the wearable under use. In contrast to the wearable system condition, the activity state should be judged by continuously sensing multiple wearables' data variation in acceleration data and the like.

B. SYSTEM CONTEXT DETECTION

To our knowledge, the context-aware scheduler is embedded and runs on the smartphone system, which could exert great influence when the software is run without system resource utilization being monitored. For example, the smartphone could crash and not respond when the preprocessed operation is executed, and CPU utilization reaches 100%. Therefore, system context detection (SCD) is well prepared to sense conditions related to the smartphone, including various system resources utilization and network states. Similar to wearable context detection, system context detection primarily senses system monitor data by using a direct mapping method. Multiple system contexts could be used in context-aware scheduling, and the system battery level for further user context detection.

C. USER CONTEXT DETECTION

In contrast to wearable context detection and system context detection, user context detection (UCD) is high level detection which integrates both wearable context and system context. In this paper, the user context is divided into three patterns: a user location pattern (ULP), a user battery charging pattern (UBCP) and a user action pattern (UAP). These three patterns describe the feature of users' behavior in terms of location, battery charging behavior and action behavior. Such user patterns could also be regarded as the influence of a user's behavior on the smartphone system.

The progress of user pattern detection is divided into two parts. The first part is user pattern classification, and the second part is user pattern calculation.

The user location pattern records the GPS coordinate location, which refers to places frequented by the user. The classification is shown in the formula (1),

$$\begin{aligned} L &= \{l_1, l_2, \ldots, l_n\} \\ S_L(t) &= l_k \end{aligned} \tag{1}$$

where each location element of l_k belongs to the location set L, and $S_L(t)$ refers to a certain location at an exact time t. The location set L could be clustered with the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) Algorithm [29]. Due to the fact that the number of items in location set can't be determined in advance, the DBSCAN is highly suitable for location clustering owing to the features it has as a density-based algorithm. Therefore, DBSCAN doesn't need to specify the number of cluster centers. As shown in formula (2), DBSCAN only needs to specify a fixed parameter ε and the minimum number of points *Minpts* to form a dense region.

$$L = DBSCAN(L^{(D)}, \varepsilon, Minpts) \tag{2}$$

The user action pattern covers basic action behaviors directly related to the wearable sensor, such as sitting, jumping, walking and running. The classification is shown in formula (3),

$$\begin{aligned} A &= \{a_1, a_2, \ldots, a_n \\ S_A(t) &= a_k \end{aligned} \tag{3}$$

where each action element a_k belongs to the user action set A, and $S_A(t)$ refers to a certain action at an exact time t.

The battery charging behavior contains two battery states as shown in formula (4),

$$S_C(t) = \begin{cases} 0 & v_{battery} \geq 0 \\ 1 & v_{battery} < 0 \end{cases} \tag{4}$$

where $S_C(t)$ refers to a certain action at a certain time t. If $v_{battery}$ is above zero, the battery charging state equals 0, which indicates that the battery is charged at time t. In contrast, if $v_{battery}$ is below zero, the battery charging state equals 1, indicating that the battery is discharged.

In order to integrate each user pattern for further pattern calculation, in this context, M refers to the user pattern set, and $\mathcal{E}_i$ is each element of M, as shown in formula (5).

$$M = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\} \tag{5}$$

After three user behavior states are classified, the user pattern calculation is carried out as follows. A user pattern contains two fundamental elements, event duration and event probability. In this context, an event could happen at a certain time. Accordingly, a user pattern aims to record the probability of the occurrence of an event and its prolonged duration at a certain time. Therefore, a user pattern could also be regarded as a user's behavior feature record.

A matrix N is mainly established to record the incidence of each event in each hour, as shown in formula (6),

$$N = \begin{bmatrix} n_{\mathcal{E}_1}^{t_1} & n_{\mathcal{E}_1}^{t_2} & \cdots & n_{\mathcal{E}_1}^{t_\mathcal{E}} \\ n_{\mathcal{E}_2}^{t_1} & n_{\mathcal{E}_2}^{t_2} & \cdots & n_{\mathcal{E}_2}^{t_\mathcal{E}} \\ \cdots & \cdots & \cdots & \cdots \\ n_{\mathcal{E}_m}^{t_1} & n_{\mathcal{E}_m}^{t_2} & \cdots & n_{\mathcal{E}_m}^{t_\mathcal{E}} \end{bmatrix} \tag{6}$$

where each element $n_{\mathcal{E}_i}^{t_j}$ corresponds to a certain time t_j with a certain event $\mathcal{E}_i$. The element count could also be expressed as C_i^j, as shown in formula (7).

$$C_i^j = n_{\mathcal{E}_i}^{t_j} \tag{7}$$

As one element of a user pattern, the duration D of a certain event ε_i is shown in formula (8), where duration is calculated by integrating the event with its time.

$$D(\varepsilon_i) = \sum_{u=1}^{C_i^j} \int_0^X S^{-1}\{\varepsilon_i\} d(\varepsilon_i) \tag{8}$$

As shown in formula (9), probability P_i^j is another element of a user pattern, which refers to the probability of element $n_{\mathcal{E}_i}^{t_j}$. It could be calculated by using the incidence of the event at a certain time C_i^j divided by the total count of all events at the same time.

$$P_i^j = \frac{C_i^j}{\sum_{k=1}^{m} C_k^j} \tag{9}$$

After event duration D and event probability P are calculated, the user pattern UP could be referred to as a set of D and P, as shown in formula (10).

$$UP = \{D, P\} \tag{10}$$

An example of user battery pattern detection is roughly described, as shown in Fig. 4. The process can be broken down into four steps. At the beginning, the wearable context detection function and system context detection function detect the wearable battery level (WBL) and system battery level (SBL), respectively. Subsequently, the battery level variation could be calculated by consistently detecting the battery level. Each irregular battery variation will be further recorded as a battery charge event. Therefore, the battery charge data set consists of both wearable battery charge events and system battery charge events. Finally, the battery charge pattern can be calculated from the battery charge data set by calculating the charging probability and charging time simultaneously.

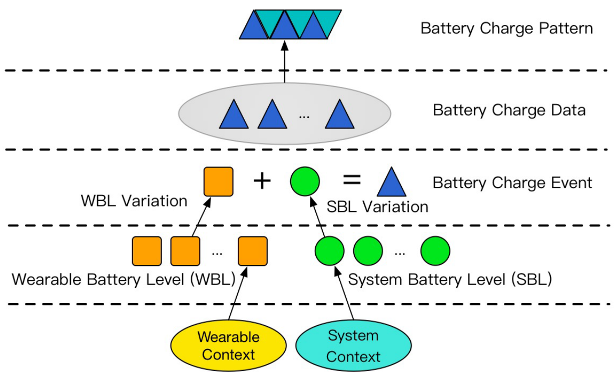

FIGURE 4. An example of battery state detection.

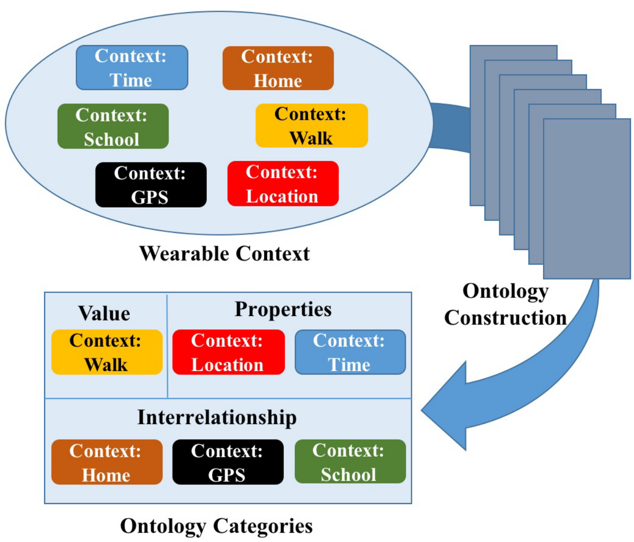

FIGURE 5. An example of ontology construction.

D. CONTEXT MANAGER

As a fundamental and indispensable part of the multiple context broker, a context manager mainly functions as storage for heterogeneous context data, as well as a context provider for ensuing processing by the context-aware scheduler. The contextual data storage structure is primarily to index related data for reading/writing. Hence, the ontology structure is well suited for contextual information management. Ontology is a specific data structure that describes the elements, attributes and related elements [30], as shown in Fig. 5.

Our system ontology consists of three categories; values, properties, and interrelationships. Such a contextual information organizing structure not only effectively limits its complexity, but also provides useful related information, according to the interrelationship category. For example, as shown in Fig. 5, the context "walk" could be recorded as a value. Subsequently, the properties store two attributes; location and time. In the interrelationship part, some related wearable contexts and system contexts are stored, including a home context, a GPS context, and a school context. The easy access to related contexts facilitates the further processes of the context-aware scheduler.

V. CONTEXT-AWARE SCHEDULER

The context-aware scheduler serves to overcome the poor adaptiveness of fixed scheduling and improve data collection based on detected context information. The context-aware engine and three context-aware sub-schedulers for data collection, preprocess and upload are illustrated respectively in the following four subsections.

A. CONTEXT-AWARE ENGINE

As described in Section III, the context-aware engine, which acts as a core part, serves to sense the contextual data for three context-aware sub-schedulers. The context-aware engine contains three different models in terms of power, resources, and data quality. These models are explained respectively in the following three subsubsections.

1) SYSTEM POWER CONSUMPTION MODEL

Since the system could be impinged upon by user behavior, modeling the system power consumption is conducive to improving the running performance of the system and extending battery life. Therefore, the system power consumption model was established to model the relationship between wearable battery consumption and data collection frequency, as well as the relationship between smartphone battery consumption and data preprocess frequency. When it comes to element input, the current battery level, the current time, the current frequency, and the user battery charging pattern will be calculated, and then the optimal scheduling frequency can be worked out. The detail of the system power consumption model is described in the next section.

2) MULTIPLE RESOURCE UTILIZATION MODEL

Similar to the system power consumption model, multiple resource aims to build a relationship between multiple system resources and system operation frequency. System operation also includes data preprocess operation and data upload operation. As for function input, the user action pattern, the current time, the current resource utilization and the network state will be calculated, and then an optimal frequency can be arrived at in terms of multiple resource utilization. The details of a model building algorithm are described in the next section.

3) DATA QUALITY OPTIMIZATION MODEL

In contrast to the previous two models, the data quality optimization model focuses on data quality. This model establishes the relationship between data collection frequency and data accuracy, and then separately establishes the relationship between data collection frequency and data efficiency. The optimal collection frequency is then calculated based on these two relationships. As for the input for the model, a wearable device set, a user location pattern, a user action pattern, and a current time will be calculated, and then the optimal frequency in terms of data quality will be arrived at. The detail of the modelling process is described in the next section.

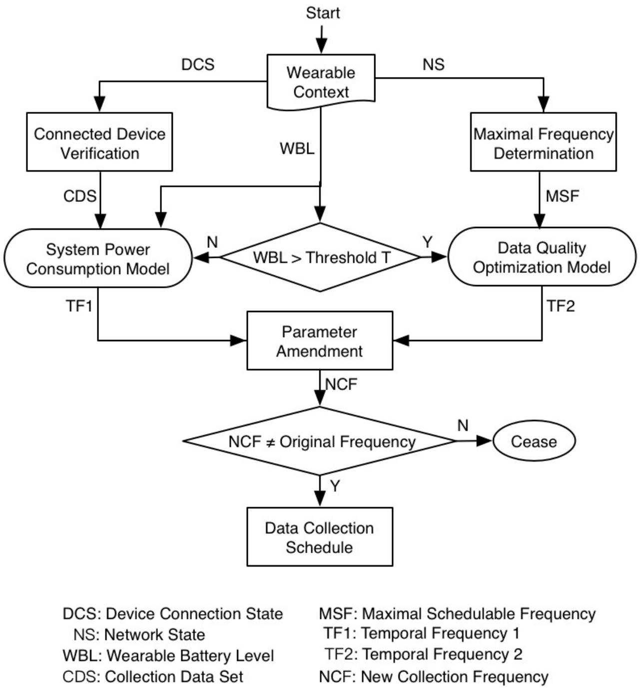

FIGURE 6. Flowchart of data collection sub-schedule.

B. DATA COLLECTION SUB-SCHEDULER

The process of the context-aware data collection sub-scheduler is illustrated in Fig. 6. In order to guarantee low scheduling consumption, the process begins when variation in the wearable context is detected.

Three wearable contexts are detected at the beginning of the data collection schedule. The first is the device connection state (DCS). DCS is a set of devices which indicate schedulable devices. Subsequently, the connection device variation is responsible for calculating the collection data set (Dc), according to DCS. The second context is the network state. The quality of the network is mainly determined by the distance between wearables and the smartphone. Such a determination of quality could be used for calculating the maximal schedulable frequency (MSF) by a maximal schedulable frequency (FM) determination function. Meanwhile, if the wearable battery level (WBL) is above the threshold T, the temporal frequency will be computed by a data quality optimization model. Otherwise, the temporal frequency will be computed by the system power consumption model. The parameter amendment function is in charge of the final frequency adjustment, according to the temporal frequency. If the modified frequency is different from the original frequency, this new schedule will correspondingly be handled by data collection.

C. DATA PREPROCESS SUB-SCHEDULER

The process of the context-aware data preprocess sub-scheduler is illustrated in Fig. 7. The process begins when changes in the system context are detected.

Similar to the processing step of a data collection sub-schedule, the data preprocess sub-schedule firstly detects the system battery level (SBL), resource state set (RSS) and preprocess amount (PA) simultaneously. Subsequently, the system measures the preprocess amount. If the amount reaches the minimal preprocess amount T, the system power consumption model will subsequently calculate the temporal frequency (TF1) by SBL. The multiple resource utilization model calculates the temporal frequency (TF2) by using the parameter RSS correspondingly. Finally, the parameter amendment function is prepared to coordinate two temporal frequencies according to RSS. For example, the strategy of the parameter amendment function is to choose the lower of the two frequencies. For instance, if TF1 is above TF2, the new preprocess frequency will be TF2.

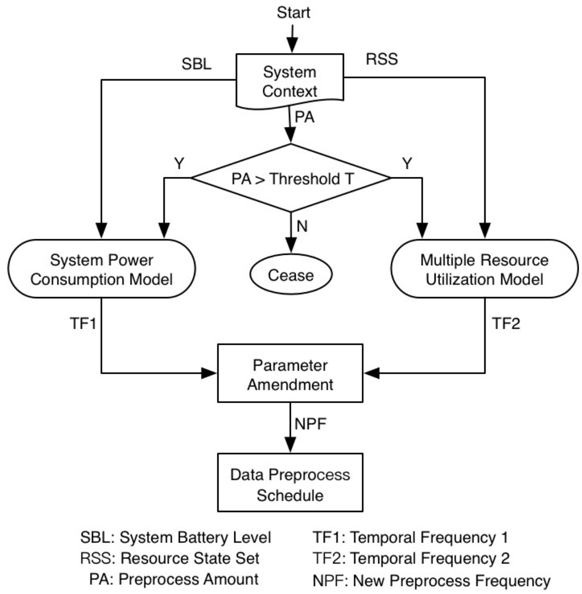

FIGURE 7. Flowchart of data preprocess sub-schedule.

D. DATA UPLOAD SUB-SCHEDULER

The process of the context-aware data uploading sub-scheduler is shown in Fig. 8. The process begins when changes in the network state are detected.

In contrast to the data collection sub-scheduler and data preprocess sub-scheduler, the data upload sub-schedule only needs to compare the upload amount with the minimal upload amount T. When this requirement is reached, the system will dispatch the network state into the multiple resource utilization model for optimal frequency calculation. In this schedule, the factor of system power consumption is not considered because battery consumption during the upload process can be ignored due to the high speed of the network and shortened upload times.

VI. MODELS IN CONTEXT-AWARE ENGINE

As described in the last section, contextual data acts as blood does in the human body, while the context-aware engine is analogous to a pumping heart that analyzes contextual information continuously. Three different models provide the

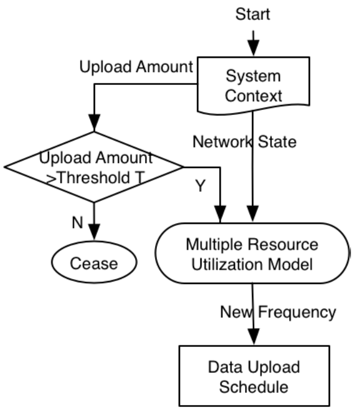

FIGURE 8. Flowchart of data upload sub-schedule.

same type of optimal frequency separately, according to different kinds of contextual data. A sub-scheduler could utilize different models under different conditions. The details of three models are illustrated respectively in the following three subsections.

A. SYSTEM POWER CONSUMPTION MODEL

As mentioned in a previous section, there are mainly three factors that influence the power consumption of the system; idle battery consumption (IBC), operation battery consumption (OBC) and user battery charging behavior. Therefore, the basic mechanism for the system power consumption model is to consider three factors at the same time.

By measuring IBC and OBC, the system total energy consumption can be easily calculated with a period of time t, as shown in formula (11),

$$E\,(t, f) = \int_{t}^{t+\frac{1}{f}} (Power_{idle} + Power_{operation}(\,f\,))dt \quad (11)$$

where the energy consumption E at each schedule interval with a certain frequency f is calculated by integrating the idle power consumption power $Power_{idle}$ and operation power consumption power $Power_{operation}(f)$.

The optimal frequency with power consumption is presented in *Procedure 1*. In this model, the user charging time t_0 is measured first according to the user battery charging pattern $UP_{bc}^{(t)}$ at a certain time t_c. The battery life t_1 can then be calculated according to formula (11). After sound preparation of necessary parameters, the system power consumption model will increase the operation frequency f continuously in order to arrive at the maximal operation frequency while guaranteeing a sufficient duration until the next charging time.

As shown in the scheduling results in Fig. 9, the red line refers to the original battery consumption with a certain operation frequency. It shows that the device is charged by the user for 12 hours, which indicates that the battery is not fully used, because the battery level is still at a high level when it is charged. Therefore, the power consumption model increases

Procedure 1 Optimal Frequency With Power Consumption

Input: Current Battery Level BL_c, Current timet_c, Current Frequency f_c, Battery Charging Pattern $UP_{bc}^{(t)}$
Output: Optimal Frequency f_0
1: $t_0 = UP_{bc}^{(t)}\{D\,(t_c)\}$
2: $t_1 = \frac{BL_c}{E(t_c,f_c)}$
3: **if** $t_1 \neq t_0$ **then**
4: $f = 1$
5: **while** $E\,(t_c, f\,) < BL_c$ **do**
6: $f = f + 1$
7: **if** $\frac{BL_c}{E(t_c,f\,)}$ **then**
8: **break**
9: **end if**
10: **end while**
11: **end if**
12: $f_0 = f$

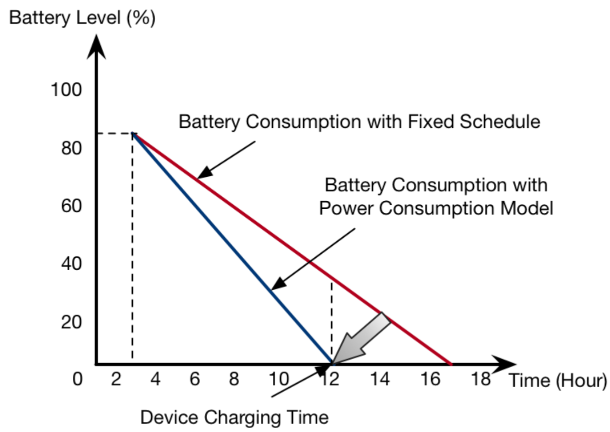

FIGURE 9. An illustration of power consumption model.

frequency to ensure the most efficient battery usage as well as a sufficient duration until the next charging time.

B. MULTIPLE RESOURCE UTILIZATION MODEL

As mentioned in the last section, the multiple resource utilization model aims to build the relationship between system resource utilization and operation frequency. This model is mainly for the data preprocess scheduler and data upload scheduler since both scheduling operations are closely related to the system resource. For example, the preprocess operation should both take into account the current CPU utilization and memory utilization. Only when both resources are satisfactory for the operation at the current frequency can this process be operated successfully. Therefore, building the relationship between each operation and multiple resource is crucial.

In this paper, the fundamental modeling element set is shown in the following formula,

$$U = \{u_a\,(f\,), u_b\,(f\,), \ldots, u_n(f)\} \quad (12)$$

where U refers to the utilization resource set, including each resource element $u_i\,(f\,)$ with a certain frequency f. According to previous research, this paper considers the relationship between frequency and resource to be linear, as shown in formula (13),

$$u_i(f) = \frac{f_i - f_i^{'}}{r_i - r_i^{'}} \tag{13}$$

where f_i refers to original operation frequency, while r_i is current resource utilization. And $f_i^{'}$ refers to the new operation frequency, while $r_i^{'}$ is new resource utilization under this frequency.

It's easy for the presented system to obtain each linear relationship between a certain frequency and a corresponding multiple utilization ratio. However, it's harder to arrange the frequency to ensure sound resource utilization. For example, if CPU utilization is 100%, but memory utilization is 60%, it's impossible to increase operation frequency due to the 'Buckets effect'. One of the basic criteria for increasing operation frequency is to ensure that utilization of all resources does not reach 100%. Therefore, in this study, two strategies to optimize multiple resource utilization are classified.

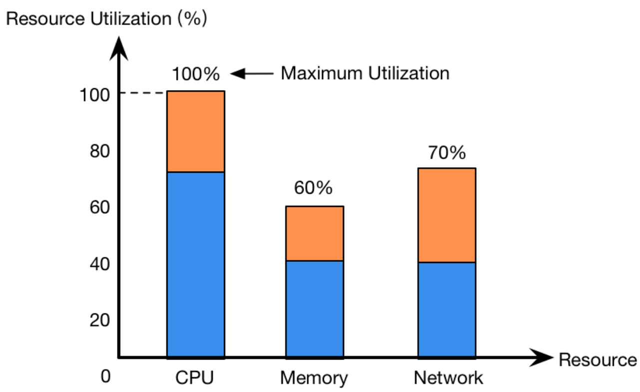

FIGURE 10. An example of maximal resource utilization strategy.

The first strategy is to ensure maximal resource utilization, as shown in Fig. 10. In this strategy, the operation frequency is increased until utilization of one resource reaches 100%. For example, the frequency keeps increasing until the system detects that the CPU has reached 100%. The advantage of this strategy is that it guarantees at least one resource is fully utilized. However, when multiple operations run at the same time, some resources may still undergo low utilization under the first strategy. That's why the second strategy is called for.

The second strategy is an idle resource balance strategy, in which each operation frequency is adjusted to reach the balance of each resource. Although a single resource may not be fully used, total usage is higher than in the former strategy, as shown in the Fig. 11.

C. DATA QUALITY OPTIMIZATION MODEL

The data quality optimization model is specialized to improve data quality. In this research, we regard data accuracy as the degree to which data integrity completely describes user behaviour, while data efficiency describes the data set with less redundancy. The data quality measuring method is shown in formula (14),

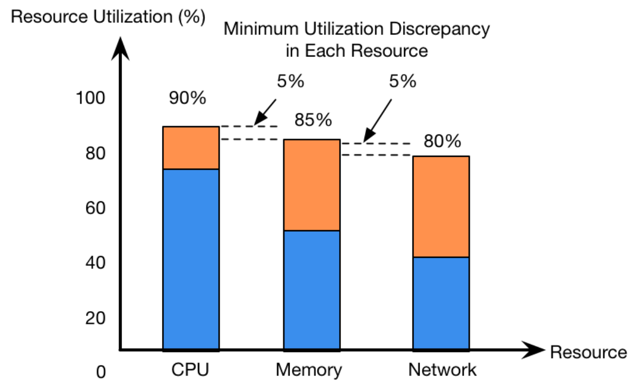

FIGURE 11. An example of idle resource balance strategy.

Procedure 2 Data Quality Optimization Model Construction

Input: Current Frequency f_c, Current Time t_c, Data Set $D(f_c)$

Output: $D_{AE}(f_c)$

1. Initialize:

$$D(f_c) = \{d_2 - d_1, d_3 - d_2, \ldots, d_n - d_{n-1} | F = F_c\}$$
$$Count_A = 0$$
$$Count_E = 0$$

2: **for** $i = 2ton$ **do**
3: **if** $(d_i - d_{i-1}) \leq \theta_{\mathbf{A}}$ **then**
4: $Count_A = Count_A + 1$
5: **end if**
6: **if** $(d_i - d_{i-1}) \leq \theta_{\mathbf{E}}$ **then**
7: $Count_E = Count_E + 1$
8: **end if**
9: **end for**
10: $D_A = \frac{Count_A}{n-1}$
11: $D_E = \frac{Count_E}{n-1}$
12: $D_{AE}(f_c) = D_A \bullet D_E$

$$D_{AE}(f) = D_A \bullet D_E \tag{14}$$

where D_A refers to the degree of data accuracy, and D_E refers to the degree of data efficiency. D_{AE} is the total degree to measure the data quality with certain f. The data quality optimization model constriction is presented in *procedure 2*. At the beginning of algorithm, the variation between each datum is calculated successively. Subsequently, $Count_A$ and $Count_E$ record the variation between two consecutive data according to two thresholds θ_A and θ_E, respectively. After the process of calculation of each data quality, D_A and D_E can be calculated according to $Count_A$ and $Count_E$. Finally, the data quality measuring degree D_{AE} can be arrived at.

With the construction of the data quality optimization model, the algorithm for optimal frequency to data quality is directed at providing the optimal data collection frequency

Procedure 3 Optimal Frequency With Data Quality Optimization Model

Input: Current time t_c, Current Frequency f_c, User pattern $UP^{(t)}$, Maximum Frequency f_M

Output: Optimal Frequency f_0

Initialize:

$$P = UP^{(t)} \{P(f_c)\}$$
$$D = UP^{(t)} \{D(f_c)\}$$
$$f = 1$$
$$f_0 = 1$$
$$D_{AE}^{Max} = 0$$

while $f \leq f_M$

  if $D_{AE}(f) > D_{AE}^{Max}$ **then**

    $f_0 = f$

  end if

end while

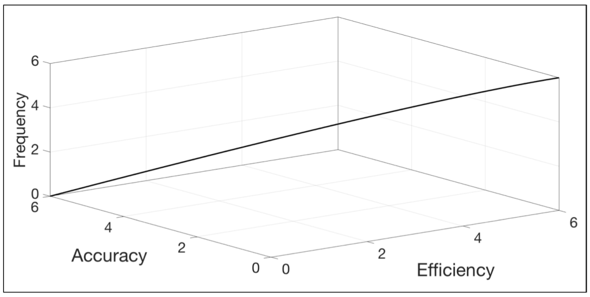

FIGURE 12. Example of data accuracy-efficiency-frequency.

according to the former constructed D_{AE} model, as presented in *procedure 3*. By changing the data collection schedule frequency, the D_{AE} is calculated. The frequency with the maximal related D_{AE} value is also obtained. This frequency is then the result of an optimal frequency according to the data accuracy and data efficiency.

An example of the data quality optimization model is shown in Fig. 12. The curve in three-dimensional coordinates reflects the data collection frequency related to data accuracy and data efficiency, respectively. According to formula (14), the data quality can be calculated by the following different frequencies. Accordingly, the curve which describes the data quality and data collection frequency is shown in Fig. 13. The peak of this curve suggests that the maximal data quality corresponds to frequency f. As a result, the system can choose the frequency f as the optimal frequency to reach the highest quality under the current context condition.

VII. EXPERIMENT AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

Our experiments contained two parts, one was the evaluation of data quality and the other the evaluation of system performance. For the first part, an iPhone6 plus was used as a

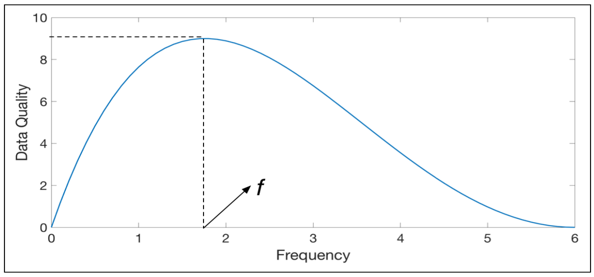

FIGURE 13. Example of data quality optimization curve.

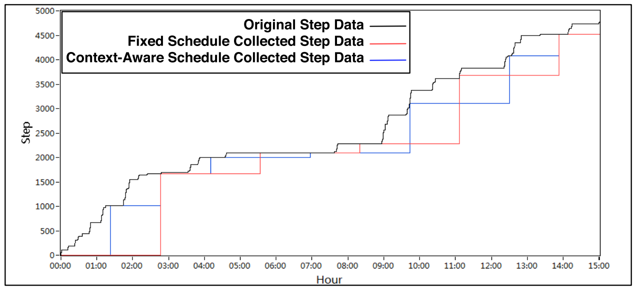

FIGURE 14. Low frequency with fixed schedule and context-aware schedule.

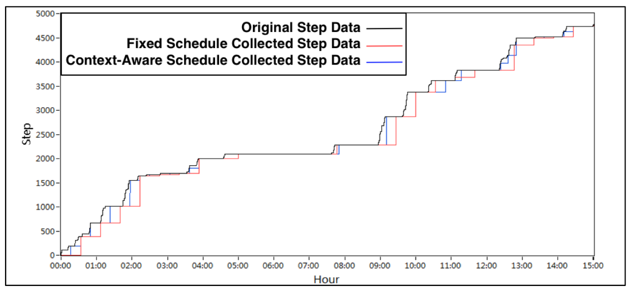

FIGURE 15. Medium frequency with fixed schedule and context-aware schedule.

client to handle data collection, data pre-process and uploads to a server. The client on the smartphone had been developed under the Mac OS X with the Swift programming language. For the second part, a simulation of a virtual smartphone model reflecting the multiple system resource utilization was implemented with multiple-programming languages, including LabVIEW 2015 and Java. Experimental charts are displayed under LabVIEW 2015 and MATLAB 2016.

B. EVALUATION OF DATA QUALITY

We conducted three groups of experiments to collect user step data under the same experimental environment. An apple watch was worn by an experimenter to provide original step data, while a fixed schedule and a context-aware schedule with a data quality optimization model were embedded into two smartphones for data collection, respectively. 15 hours of step data was recorded subject to two different schedules, as shown in Fig. 14, Fig. 15, and Fig. 16.

We mainly classified the frequency into three frequencies. That is low frequency, medium frequency and high frequency. Three figures draw the three differing scheduling frequencies

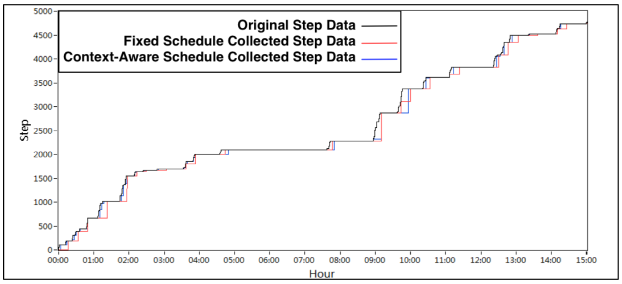

FIGURE 16. High frequency with fixed schedule and context-aware schedule.

TABLE 1. Experimental data with fixed schedule.

Frequency	Data Accuracy	Data Efficiency	Accuracy-Efficiency	Fitting
Low	2	4	8	72.9%
Medium	8	18	144	93.3%
High	29	24	696	96.4%

TABLE 2. Experimental data with context-aware schedule.

Frequency	Data Accuracy	Data Efficiency	Accuracy-Efficiency	Fitting
Low	3	6	18	84.0%
Medium	20	22	440	95.8%
High	106	42	4452	97.4%

under the same step data set. The black curve is the original step data, while the red curve illustrates the step data collected under a fixed schedule. A blue curve also shows the step data collected by a context-aware schedule. The experimental data under two schedules is shown in Table 1 and Table 2 respectively.

According to the experiments, four evaluable factors can be calculated for data quality evaluation, namely data accuracy, referring to the number of data which are equal to the original data, data efficiency, referring to the number of redundant data, data quality optimization, referring to the combination of data accuracy and data efficiency, and the quality of the fit, referring to the degree of similarity between the scheduling data curve and original data curve. The result shows that the data collected by context-aware scheduler is overwhelmingly better than data collected by fixed scheduler according to each factor. These two set of results also show that, in a certain frequency range, the higher frequency scheduled, the better performance of data quality is.

C. EVALUATION OF SYSTEM PERFORMANCE

We conduct two groups of 15 hours' experiments with different smartphone resource utilization at a virtual Environment. The simulation of smartphone resources contains CPU, memory and battery level. The first group of experiments was four kinds of data processing experiments which show

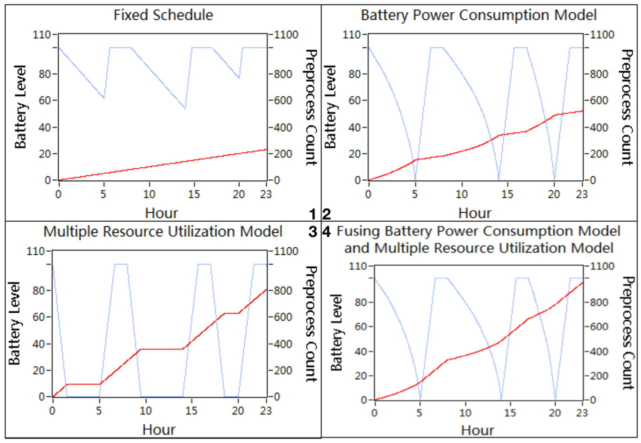

FIGURE 17. Data preprocess with four kinds of scheduling models under idle resource utilization.

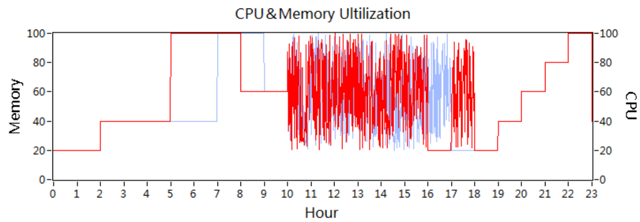

FIGURE 18. Dynamic CPU and memory utilization.

the scheduler guided by four kinds of scheduling models under the idle resource utilization, and the second part of experiment is operated under dynamic resource utilization.

In Fig. 17, the red curve refers to the total preprocess counts, while the blue curve shows the variation in battery level. During the first experiment, the system idle resource utilization stayed at a constant utilization, whereas the battery separately charged at 5 hours, 14 hours and 20 hours precisely. It's clear that the highest preprocess counts are 980, which were scheduled by fusing the battery consumption model with the multiple resource utilization model, while the lowest preprocess counts are 210, which were scheduled by the fixed scheduler. In addition, the scheduler with the battery consumption model and the scheduler with the multiple resource utilization model are both better than the fixed scheduler due to the dynamic frequency changing strategy based on the charging time.

The second group of experiments were conducted under the dynamic CPU and Memory utilization as shown in Fig. 18. After the first 5 hours, both CPU and Memory are in low usage, which means there is no extra app usage. At 5 to 10 hours, resources are in high usage, which means the user is using the smartphone during this period. Then, some random apps ran from 10 to 18 hours. Finally, both resources increased from low utilization to high utilization.

Fig. 19. shows the data preprocess operation scheduled under four kinds of model. The result is similar to the experiments conducted in the idle resource utilization. The best preprocess performance was scheduled by fusing the battery

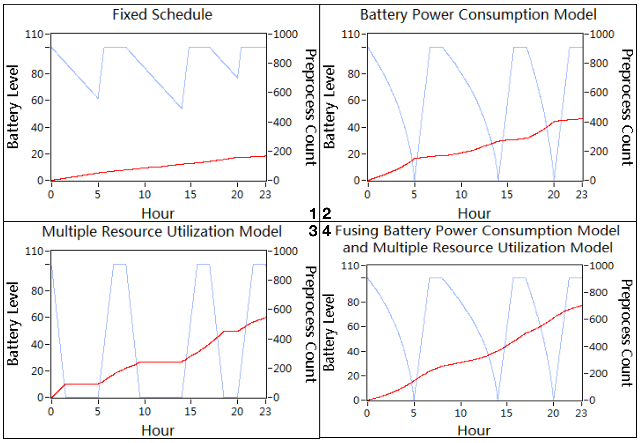

FIGURE 19. **Data preprocess with four kinds of scheduling models under dynamic resource utilization.**

power consumption model and multiple resource utilization model, while the worst performance was conducted subject to a fixed schedule. Regarding the two aspects of the experiments, we can also the draw the conclusion that the performance of multiple resource utilization is based on the battery charging pattern. This means the best schedule could also be the best multiple resource utilization, if the battery continues charging or the resource utilization can be neglected.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a context-aware scheduler for three main objectives. The first objective is to automatically collect user personal data from multiple wearable devices. Due to the popularity and compatibility of multiple wearables, a smartphone was chosen as a terminal for data collection as well as data preprocess and upload. The second is to fully take advantage of contexts which may influence data collection, preprocess and upload. To do so, a context broker was proposed and implemented accordingly. By sensing the wearable context and system context, the context broker was able to provide guidance for further adjusting scheduling parameters to reach a generally acceptable level system performance for practical use. Considering the influence of user behavior, an extra user context was detected according to wearable context and system context. Three user patterns were detected to measure data quality as well as set further scheduling, namely a user location pattern, a user action pattern and a battery charge pattern. In addition to the context detection, a context manager inside the context broker was implemented based on an ontology for contextual information storage and provision. The third objective is to reach the optimal schedule in terms of data quality and system performance. A context-aware engine was proposed for modeling contextual data under three different models, namely a data quality optimization model, a system battery consumption model and a multiple resource utilization model.

However, much work remains for future study into two specific aspects. First, more contextual factors should be taken into consideration for the context-aware scheduler, especially the user context. Secondly, models from the context engine should be refined for more precise context estimation. Third, more experiments will have to be carried out to further evaluate and improve data collection, preprocess and upload in the system.

REFERENCES

[1] J. Ma, L. T. Yang, B. O. Apduhan, R. Huang, L. Barolli, and M. Takizawa, "Towards a smart world and ubiquitous intelligence: A walkthrough from smart things to smart hyperspaces and UbicKids," *Int. J. Pervasive Comput. Commun.*, vol. 1, no. 1, pp. 53–68, 2005.

[2] J. Ma, J. Wen, R. Huang, and B. Huang, "Cyber-individual meets brain informatics," *IEEE Intell. Syst.*, vol. 26, no. 5, pp. 30–37, Sep./Oct. 2011.

[3] B. Guo, Z. Yu, and X. Zhou, "A data-centric framework for cyber-physical-social systems," *IT Prof.*, vol. 17, no. 6, pp. 4–7, Nov./Dec. 2015.

[4] L. A. Tang, J. Han, and G. Jiang, "Mining sensor data in cyber-physical systems," *Tsinghua Sci. Technol.*, vol. 19, no. 3, pp. 225–234, 2014.

[5] W. Jie, M. Kai, W. Furong, H. Benxiong, and M. Jianhua, "Cyber-I: Vision of the individual's counterpart on cyberspace," in *Proc. IEEE Int. Conf. Depend. Autonomic Secure Comput. (DASC)*, Dec. 2009, pp. 295–302.

[6] J. Ma and R. Huang, "Wear-I: A new paradigm in wearable computing," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Depend. Autonomic Secure Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, Oct. 2015, pp. 1063–1068.

[7] A. Vinciarelli and G. Mohammadi, "A survey of personality computing," *IEEE Trans. Affect. Comput.*, vol. 5, no. 3, pp. 273–291, Mar. 2014.

[8] A. Guo and J. Ma, "A smartphone-based system for personal data management and personality analysis," in *Proc. 13th IEEE Int. Conf. Pervasive Intell. Comput. (PICom)*, Oct. 2015, pp. 2114–2122.

[9] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.

[10] T. Hashimoto, J. R. Stedinger, and D. P. Loucks, "Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation," *Water Resour. Res.*, vol. 18, no. 1, pp. 14–20, 1982.

[11] J. Alejandro, N. Sebe, and D. Gatica-Perez, "Human-centered computing: A multimedia perspective," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, Oct. 2006, pp. 855–864.

[12] M. Billinghurst and T. Starner, "Wearable devices: New ways to manage information," *Computer*, vol. 32, no. 1, pp. 57–64, Jan. 1999.

[13] B. C. Singh, B. Carminati, and E. Ferrari, "A risk-benefit driven architecture for personal data release," in *Proc. IEEE 17th Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2016, pp. 40–49.

[14] X. Zhou, B. Wu, Q. Jin, W. Li, and N. Y. Yen, "Personal data analytics to facilitate cyber individual modeling," in *Proc. IEEE 14th Conf Depend. Autonomic Secure Comput*, Aug. 2016, pp. 39–46.

[15] E. Aïmeur and M. Lafond, "The scourge of Internet personal data collection," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2013, pp. 821–828.

[16] B. N. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proc. IEEE Workshop Mobile Comput. Syst. Appl.*, Dec. 1994, pp. 8–9.

[17] W. Liu, X. Li, and D. Huang, "A survey on context awareness," in *Proc. Int. Conf. Comput. Sci. Service Syst. (CSSS)*, Jun. 2011, pp. 144–147.

[18] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[19] Y. Lee, S. Kang, C. Min, Y. Ju, I. Hwang, and J. Song, "CoMon+: A cooperative context monitoring system for multi-device personal sensing environments," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1908–1924, Aug. 2015.

[20] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2014, pp. 68–81.

[21] T. Tran, "Context-aware interflow network coding and scheduling in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9299–9318, Nov. 2016.

[22] T. Mavridis, L. Petrillo, J. Sarrazin, A. Benlarbi-Delaï, and P. De Doncker, "Human influence on 60 GHz communication in close-to-user scenario," in *Proc. 14th URSI General Assembly Sci. Symp. (URSI GASS)*, Aug. 2014, pp. 1–4.
[23] N. Matsumura and Y. Sasaki, "Understanding leadership behavior in human influence network," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2006, pp. 95–102.
[24] D. Dzvonyar, S. Krusche, R. Alkadhi, and B. Bruegge, "Context-aware user feedback in continuous software evolution," in *Proc. IEEE/ACM Int. Workshop Continuous Softw. Evol. Delivery (CSED)*, May 2016, pp. 12–18.
[25] E. S. Marin and C. L. de Carvalho, "Search in social networks: Designing models and algorithms that maximize human influence," in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, Jan. 2014, pp. 1586–1595.
[26] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 299–318, 2008.
[27] T. S. Andre, H. R. Hartson, S. M. Belz, and F. A. McCreary, "The user action framework: A reliable foundation for usability engineering support tools," *Int. J. Human-Comput. Stud.*, vol. 54, no. 1, pp. 107–136, 2001.
[28] A. C. Baughman and M. Ferdowsi, "Double-tiered switched-capacitor battery charge equalization technique," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2277–2285, Jun. 2008.
[29] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.
[30] S. Bechhofer, "OWL: Web ontology language," in *Encyclopedia of Database Systems*. New York, NY, USA: Springer, 2009, pp. 2008–2009.

AO GUO received the M.E. degree from the Huazhong University of Science and Technology, China, in 2016. He is currently pursuing the Ph.D. degree with Hosei University, Japan. His research interests lie in the intersection of human-computer interaction and ubiquitous computing, including context-aware data collection system and personality analysis. He received the best paper award at CPSCom 2016.

JIANHUA MA is currently a Professor with the Department of Digital Media, Faculty of Computer and Information Sciences, Hosei University, Japan. He had 15 years of working experience with the National University of Defense Technology, Xidian University, and with the University of Aizu, Japan. He has authored over 200 papers, and edited over 20 books/proceedings and over 20 journal special issues. His research interests include multimedia, networks, ubiquitous computing, social computing, and cyber intelligence. He is the Co-Founder of the IEEE International Conference on Ubiquitous Intelligence and Computing, the IEEE Conference on Cyber, Physical and Social Computing, and the IEEE Conference on Internet of Things.