File Annotation and Sharing on Low-End Mobile Devices

Sadaqat Jan, Maozhen Li, Ghaidaa Al-Sultany and Hamed Al-Raweshidy
School of Engineering and Design
Brunel University, Uxbridge, UB8 3PH, UK
Sadaqat.Jan@brunel.ac.uk

Abstract— **Rapid technological evolutions in mobile industry have revealed new challenges for scientists and researchers by empowering low-end devices in the last decade. Mobile phones with excellent computing power and added features especially with huge storage capacities are very common in the market. As the storage capacity grows on such a device, it becomes very tedious for users to manage and organize different type of files. The file system of such devices itself provide some managing mechanism to store same type of files separately however it proves scarce as the number of files grows. Furthermore, limited user interface on these devices makes file retrieval more difficult. This paper presents detailed analysis of meta-data significance in our proposed framework to annotate and search files on low-end mobile devices. File attributes are extracted from the underlying operating system of the device and used as meta-data of the corresponding file. The meta-data is then used for searching the required file. The file search can be performed on the device itself or on other devices when connected in a Personal Area Network (PAN) through Bluetooth. The proposed framework is implemented in Java Micro Edition and a full featured application is developed to demonstrate its validity. The application provides annotation, search options, Bluetooth module, file sharing and transfer options. Results from various performance tests show significant improvement in searching a required file through our proposed framework.**

Keywords- J2ME, File Annotation, Mobile Devices, kXML.

I. INTRODUCTION

Importance of mobile phones cannot be neglected which plays a vital role to cope with our daily life [1]. As the technology advances, mobile phones are now commonly equipped with larger storage capacity and added features like navigation system, camera and additional communication interfaces. However problem arises once user starts using such features and it generates a large number of files. These stored files may include images or video files captured from mobile camera, downloaded or any other type of files. As the number of files grows, it becomes intricate to remember the file name and its contents. The device's file system provide some basic mechanism to store same type of file in separate directories but it also does not help because of its naming convention used. For example, mobile-phone camera application will store images as *Image001*, *Image002* and so on, which are not descriptive enough to remember. In addition, user may store other types of files like *.pdf*, *.doc* or *.ppt*. On the other hand, information search have been a challenging chore for researchers as its volume and dimension grows up whether it is on the internet or personal computing device [2]. In this paper, we present a framework to annotate files on low-end devices like mobile phones, Personal Digital Assistants (PDAs) and other Mobile Information Device Profile (MIDP) compliant devices. The meta-data analyses are also presented to show its significance while searching a required file using the proposed framework.

Meta-data is extracted from the underlying file system of the device and each file is decorated with its attributes. The meta-data is subsequently stored locally in Extensible Mark-up Language (XML) format, which means that the framework is not contingent to communication medium for neither storage nor retrieval of meta-data. Annotation process automatically annotates each file with three basic attributes however two optional tags can also added. A compact version of XML is used in order to parse the XML data in search module of the framework. File can be searched automatically through any of the attributes on device itself or even on other connected devices. The meta-data of all annotated files can also be viewed as a complete list. Performance evaluation shows that the parser does not degrade the search performance in the framework. Bluetooth module is also added to support search in Personal Area Network (PAN) environment. Files can be shared or transferred to other connected devices through Bluetooth. All modules are implemented in most widely used platform Java Micro Edition (J2ME) [3, 4].

The remainder of the paper is organized as follows. Section-II provides an overview of existing related research. J2ME platform is briefly explained in Section-III while Section-IV presents the proposed framework. Results and analysis are discussed in Section-V. Finally Section-VI concludes the paper with an outline for future extension to this work.

II. RELATED RESEARCH

Recent research indicates some diverse techniques to handle file management and its efficient retrieval [5]. These efforts are also much needed on handheld and resource-limited devices such as PDAs and mobile phones. Various techniques and research directions have been explored recently including semantic aware file system which is one of the most agreed alternatives to traditional hierarchical file system and extended in [6]. The data model was proposed to represent semantic

978-1-4244-5934-6/10/$26.00 ©2010 IEEE

information in file systems. Its two main features are extensibility to cover semantic information and handling schema evolution. Similarly, the semantic file system approach [7] increased the file scope of their namespace by assigning two generic pairings (category and value). Various procedures were proposed for integrating attribute based naming with orthodox hierarchal system in [8,9,10,11]. Efforts have been made to extend file annotation on handheld devices but proved much complicated due to resource constraints. However, a number of different approaches were presented to annotated images on mobile phones. The images were annotated with contextual information extracted from other devices connected in PAN environment [12].

In Photo annotation [13] the images are annotated with mobile phone ID, username, date and time. Web services and ontology based image annotation were presented in the framework [14,15] to annotate files. Any proposed framework, using a network medium to store or retrieve meta-data, makes it network dependent. In such mechanism, either annotation process or retrieval process will not be viable in case of network failure. Content based search is another choice of scientists to tackle the issue however such efforts will be computationally expansive for low-end devices. No such real efforts can be seen to annotate all type of files on low-end devices.

III. JAVA 2 MICRO EDITION (J2ME)

J2ME is one of the most commonly used powerful platforms for mobile phone application development. This platform was developed by Sun Microsystems in 1999 which offers a robust environment for application developers [16, 17]. Java Community Process (JCP) is process to develop standards for Java technology and they defined various configurations and profiles to fulfill diverse requirements of application developers [18]. J2ME uses a compact version Kilo Virtual Machine (KVM) instead of full Java Virtual Machine (JVM) to make the whole architecture more modular and scalable. Configuration is the run-time environment and classes operating on a device while profile is the set of domain specific classes to implement relevant features on a related group of low end devices. As shown in Fig. 1, configuration layer has two parts namely, Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). The standardized specifications for both configurations are defined in JSR-139 [19].

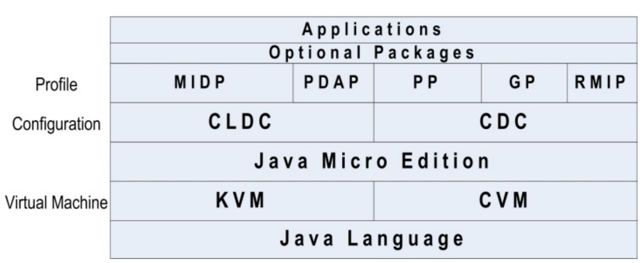

Fig. 1 J2ME Architecture

The Profile layer consists of five different profiles, in which Mobile Information Device Profile (MIDP) is the most commonly used profile and its standards are defined in [20]. MIDP profile is usually used with CLDC as it contains classes to deal with networking, storage and user interface capabilities. MIDlets are MIDP applications and the software that implements MIDP running in KVM. We used the same combination to implement our proposed framework using the core classes for file system, Bluetooth and graphical user interface (GUI). The meta-data is needed to be processed and manipulated in XML format but its parser tends to be bulky for resource limited devices. XML is a meta-markup language which was endorsed by W3C [21] and has become universally accepted specification for exchanging documents and data across applications and platforms [22]. We used kXML to parse XML data, which is a lighter and compact version of XML parser specially designed for low-end devices and is exclusively used on J2ME platform [23]. The mainly used three types of parsers includes model, push and pull parser. Model parser reads the complete document and then creates a presentation in memory and sometime requires huge memory depending on the size of the XML document [24]. Push parsers process data definitions before the creation of tree structure in memory, while pull parser reads the data first before it starts parsing. Pull parser uses recursive functions to create document tree and hence are more suitable for handheld devices.

IV. IMPLEMENTATION

A complete working model of the proposed framework is implemented in J2ME and consists of several modules to perform different functions as shown in Fig. 2. An application scenario is also presented to illustrate the complete working model. Annotation module automatically annotates files with basic file attributes and stores annotation in XML format. Search module at the top, is capable of searching the required file with various options including the view option to read the whole XML file at once. Bluetooth module is included to support search in PAN environment and to transfer file between connected devices. Details of each module are given in subsequent sections.

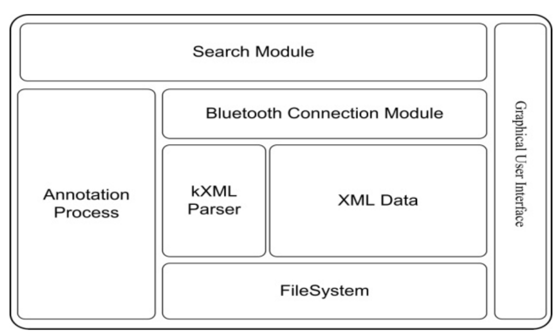

Fig. 2 Proposed Framework Architecture

A. Application Scenario

In this scenario, a user took a few group pictures on his birthday party using his mobile phone camera which were saved as application default names *image212* and *image213.* One year later, user intends to view the pictures but forgot the names of the required files. He is also not sure whose mobile phone was used, because most of his family members took pictures at that occasion. To search for the required picture he has to browse all image files, opening all images till he finds the required one. He has to repeat the same browsing process on mobile phones of other family members. In similar situations, the framework presented in this paper can be used to easily retrieve the required file by using its advanced search options. User has to enter his birth date and can retrieve the files created on that date, even if he did not annotate that photo with optional tags. Similarly, if the required file is not on his mobile phone, it can be searched on all connected mobile phones at once.

B. Annotation Module

This module interacts with the underlying operating system of device to haul out vital attributes of files. These attributes are used as annotation tags for each corresponding file. In other words, each file is annotated with its own attributes which are extracted from file system. All attributes are parsed and stored locally in XML format. The meta-data consists of two parts namely, *Automated* and *Optional*. In *Automated* meta-data part, files are annotated automatically with three basic attributes which includes file-name, file-size and date-of-creation, while two additional tags can be appended through *Optional* meta-data part. These two optional tags are *Keyword* and *Description,* which users have to type in for each file. Fig.3 is presented to explain the annotation process. Annotation is a one time process but it can be edited or updated anytime for any stored file on the device. The implemented MIDlet's main screen is shown in Fig. 4 while Fig. 5 shows the list of files to be annotated, meta-data for all annotated file can also be viewed as a single list as shown in Fig. 6.

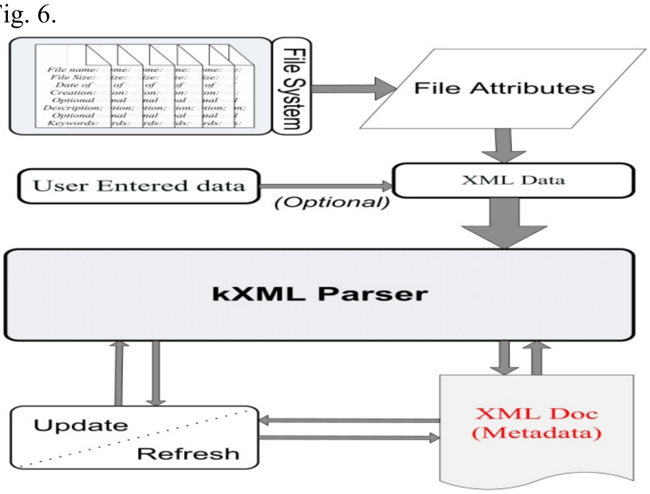

Fig. 3 Annotation Process

The process of updating meta-data for a single file named *"nature.png",* is shown in Fig. 7 as an example, while the updated list can be seen in Fig. 8.

C. Search Module

Search module plays vital role in proposed framework and interacts with nearly all other modules included. This module uses kXML parser for processing XML document in order to search for information about the required file. As mentioned in the previous section that files are annotated with three automatic and two optional tags, search can be performed with different options using available attributes as shown in Fig. 9, however the two *optional tags* can only be used as search option if that file is annotated with the *optional tags*. If search is intended in a PAN environment, search module uses Bluetooth module to send search query to other connected devices. All query receiving devices also parse their local XML documents and send back the search results. Once the file name is found, it can be shared and send to requesting device. Fig. 10 shows the attributes which are stored in XML file as meta-data.

Fig. 4 Main MIDlet

Fig. 5 Annotation Files list

Fig. 6 Meta-data List

Fig. 7 File Annotation

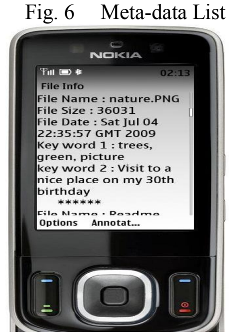

Fig. 8 Meta-data Updated List

Fig. 9 Search Options

D. Bluetooth Module

Bluetooth is a short range, low power and low cost radio communication and it is available in the majority of new handheld devices. In our proposed framework, this module is used to send search queries and query results as shown in Fig. 11. This module also provides functionalities for sharing, un-sharing and transferring files between devices. The Bluetooth protocol stack is standardized in JSR-82 which gives good control for stack initialization, device management, device discovery, service discovery, and communication. Service Discovery Protocol (SDP) is used to discover the nearby devices and files shared by other users in the network.

```
<? xml version="1.0" ?>
- <start>
  - <File>
      <FileName>nature.PNG</FileName>
      <FileSize>36031</FileSize>
      <FileCDate>Sat July 04 22:35:57 GMT 2009</FileCDate>
      <KeyWord>trees, green ,picture</KeyWord>
      <Description>Visit to a nice place on my 30th
           Birthday with my family and friends </Description>
    </File>
  - <File>
      <FileName>classnote.doc </FileName>
      <FileSize>278</FileSize>
      <FileCDate>Mon Jul 06 11:06:15 GMT 2009 </FileCDate>
      <KeyWord>J2ME, WNCC, notes </KeyWord>
      <Description>Lecture notes</Description>
    </File>
</start>
```

Fig. 10 Annotation Tags

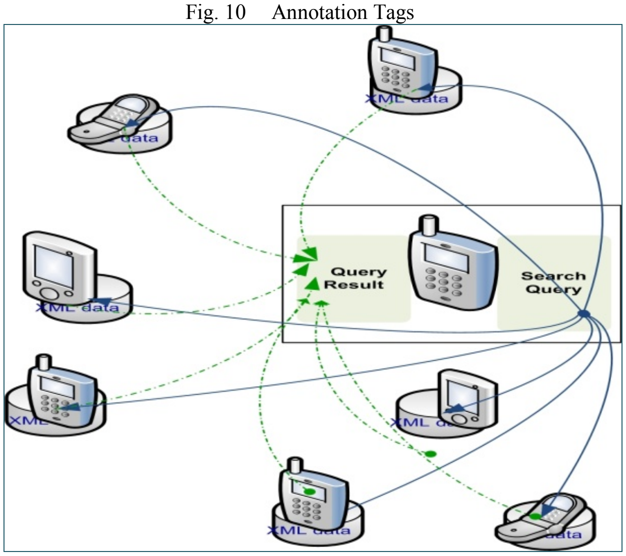

Fig. 11 File Search in PAN Environment

V. RESULTS AND ANLAYSIS

To evaluate the significance of meta-data in our proposed framework, we formulated two set of tests comprising of 25 search queries. For the first set, all files were automatically annotated and the optional tags were also added, however for the second test, files were only annotated automatically and additional tags were left empty. Results shown in Table-1 indicate that success rate is 84% for the set which was fully annotated and 72% for the second set.

TABLE I. SUCCESSES AND QUERY TYPES

No of queries	Query types	Success
25	File Name, File size, Date of creation, keywords and description.	21
25	File Name, File size and Date of creation	18

The significance of user-typed additional tags can be measured by the difference between two sets of results, which is 12% in this case. We can use probabilistic evaluation to generalize this implication. A file search query may have two possible outcomes either success or failure, therefore, based on value of p and q, a generalized model is presented to quantify the efficiency of search mechanism based on Binomial Distribution $b(x; n, p)$.

$$b(x;n \qquad (1)$$

The Binomial Distribution gives good approximation for probability of x successes for each system as the number of trial n increases. The successes probability can be denoted by P for the first set and P_a for the second set (which is automated annotation only). When the number of successes x is varied keeping the number of trails n constant i.e. $n = 100$, the comparison of P and P_a is given in Table II.

A probability comparison of getting x successes for a number of trials for both sets of test is shown in Fig. 12. The graph shows that the probability of getting a successful result is higher for P which is 70% and above as compared to P_a. It is more likely to get a successful result if files are annotated with the optional tags along with *automated* meta-data tags. By using maximum likelihood estimator we can calculate the *first* expected success in both cases and can compare the results. In order to calculate the parameter estimation for geometric distribution of our proposed search mechanism we first get the likelihood function for a random sample x_1, x_2, $x_3 \ldots x_n$. The likelihood function is given by:

$$L(p) = (1-p)^{x_1-1}\, p\,(1-p)^{x_2-1} \cdots (1-p)^{x_n-1}\, p$$
$$= p^n (1-p)^{\sum x_i - n},\ (0 \le p \le 1)$$

taking the natural logarithm

$$= \ln L(p) = n\ln(p) + \left(\sum_{i=1}^{n} x_i - n\right)\ln(1-p), \qquad (0 < p < 1)$$

taking the derivative with respect to

$$\frac{d \ln L(p)}{dp} = \frac{n}{p} - \frac{\sum_{i=1}^{n}(x_i - n)}{1-p} = 0$$

after solving for p, we get:

$$p = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\bar{x}}$$

TABLE II. COMPARISON OF CALCULATED BINOMIAL DISTRIBUTION

x	50	60	70	80	90
P=0.84	2.65E-15	5.75E-09	1.95E-04	5.67E-02	2.92E-02
P_a=0.72	1.69E-06	0.0029137	0.07869	0.01815	7.41E-06

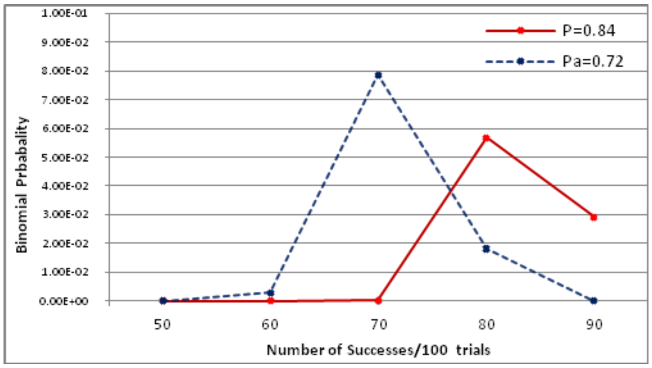

Fig. 12 Probability Comparison

The maximum likelihood estimator of is:

$$\hat{p} = \frac{n}{\sum_{i=1}^{n} x_i}$$

the estimate of p is the number of successes divided by the number of trials

$\hat{p} = \frac{1}{x}$ or $x = \frac{1}{\hat{p}}$

where x gives us the number of trials in which the first successes occurs. By calculating for both sets, the value of p for the *automated* annotation and *optional* annotation is 0.84 and 0.72 respectively, and the value of x is 1.190 and 1.388. It can clearly be concluded that if files are tagged with automated meta-data only, user will need 1.388 trials to get the required file and 1.19 trials will be required if files are tagged with optional tags along with automated metadata.

VI. CONCLUSION AND FUTURE WORK.

This paper presents a framework to annotate and share files on resource limited handheld devices. The framework is implemented as full featured MIDlet and results commend the proposed framework's importance. The significance of adding *optional tags* as meta-data is also analyzed and result shows that it increases the search ability by approximately 14%. A file is searched by matching the search query with the stored meta-data which means that any of the exact field from meta-data is required to get the search successful. As an extension to our current work, we plan to tackle this issue by adding ontology to the stored meta-data to add the semantic based search to the existing framework. For this purpose, a framework is needed to provide programmatic environment to support Ontology Web Language and Resource Description Framework (OWL/RDF) along with computationally lighter reasoner for resource limited devices. Various techniques will also be analyzed to create and update ontology and its role in file search.

REFERENCES:

[1] http://www.marketresearchworld.net/index.php?option=com_content&task=view&id=2748&Itemid=77

[2] Girgensohn, A., Adcock, J., Cooper, M., Foote, J. and Wilcox, L. Simplifying the Management of Large Photo Collections. In Proc.INTERACT '03. IOS Press (2003).

[3] G. Lawton, "Moving Java into Mobile Phones," Computer, vol. 35, no. 6, June 2002, pp. 17–20.

[4] D. S. Kochnev and A. A. Terekhov. Surviving Java for Mobiles. IEEE Pervasive Computing, 2(2):90–95,2003.

[5] Soules CAN, Ganger GR (2003) Why can't I find my files? New methods for automating attribute assignment. In: Proceedings of HotOS IX: the 9th workshop on hot topics in operating systems. USENIX Association

[6] Xu, Z., Karlsson, M., Tang, C., and Karamanolis C. "Towards a Semantic-Aware File Store." 9th Workshop on Hot Topics in Operating Systems (HotOS IX). May 18-21, 2003.

[7] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O'Toole Jr. Semantic file systems. ACM Symposium on Operating System Principles. Published as Operating Systems Review, 25(5):16– 25, 13–16 October 1991.

[8] B. Gopal and U. Manber. Integrating content-based access mechanisms with hierarchical file systems. Symposium on Operating Systems Design and Implementation, pages 265–278. ACM, 1999

[9] S. Sechrest and M. McClennen. Blending hierarchical and attribute-based file naming. International Conference on Distributed Computing Systems, pages 572–580, 1992

[10] Lalis S, Karypidis A, Savidis A (2005) Ad-hoc composition in wearable and mobile computing. Commun ACM 48(3):67–68

[11] Lalis S, Karypidis A, Savidis A, Stephanidis C (2003) Runtime support for a dynamically composable and adaptive wearable system. In: Proceedings of the 7th IEEE international symposium on wearable computing, pp 18–21

[12] A. Karypidis and S. Lalis. (2007)Automated context aggregation and file annotation for PAN-based computing. Personal and Ubiquitous Computing, 11(1):33–44

[13] Anita Wilhelm , Yuri Takhteyev , Risto Sarvas , Nancy Van House , Marc Davis, (2004)Photo annotation on a camera phone, CHI '04 extended abstracts on Human factors in computing systems, Vienna, Austria

[14] Monaghan, F., O'Sullivan, D (2006) Automating Photo Annotation using Services and Ontologies, Proceedings of Mobile Services and Ontologies Workshop

[15] G. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga,(2001) Ontology-Based Photo Annotation, IEEE Intelligent Systems, vol. 16, pp. 66-74

[16] Sun Developers Network (SDN) Java Micro Edition; http://java.sun.com/javame/index.jsp.

[17] James White, (2001) An introduction to Java 2 micro edition (J2ME); Java in small things Proceedings of the 23rd International Conference on Software Engineering, p.724-725 Toronto, Ontario, Canada

[18] Community Development of Java Technology Specification (JCP); http://jcp.org/en/home/index

[19] Connected limited device configuration (CLDC) specs. JSR 139, http://jcp.org/aboutJava/communityprocess/final/jsr139/index.html

[20] Java Specification Requests, Mobile Information Device Profile (MIDP) JSR 118, http://jcp.org/en/jsr/detail?id=118

[21] W3C: http://www.w3.org/

[22] Extensible Markup Language, http://www.w3.org/XML/

[23] kXML, http://kxmlrpc.objectweb.org/

[24] Parsing XML in J2ME, by Jonathan Knudsen, http://developers.sun.com/techtopics/mobility/midp/articles/parsingxml/