

# SMMART, a Context-Aware Mobile Marketing Application: Experiences and Lessons

Stan Kurkovsky<sup>1</sup>, Vladimir Zanev<sup>2</sup>, and Anatoly Kurkovsky<sup>3</sup>

<sup>1</sup> Computer Science Department,  
Central Connecticut State University,  
1615 Stanley Street, New Britain, CT 06050, USA  
KurkovskySta@ccsu.edu

<sup>2</sup> Department of Computer Science,  
Columbus State University,

4225 University Avenue, Columbus, GA 31906, USA  
Zanev\_Vladimir@colstate.edu

<sup>3</sup> Department of Mathematics, Physics and Computer Science,  
University of the Sciences in Philadelphia,  
600 South Forty-third Street, Philadelphia, PA 19104, USA  
a.kurkov@usip.edu

**Abstract.** A new class of m-commerce applications is emerging due to the unique features of handheld devices, such as mobility, personalization and location-awareness. This paper presents SMMART, a context-aware, adaptive and personalized m-commerce application designed to deliver targeted promotions to the users of mobile devices. SMMART adapts to changing interests of its user by monitoring his or her shopping habits and guarantees the privacy of its users by not transmitting any personally identifiable information to the retailers. We describe our experiences of building and evaluating a fully functional prototype of SMMART implemented for Pocket PCs.

## 1 Introduction

M-commerce is a branch of electronic commerce, in which mobile devices and their network connection medium are used in the process of buying and selling of services, and products. Wireless mobile devices possess unique features: ubiquity (they are affordable and portable), personalization (a device belongs to and can be identified with a single individual), and location awareness (a wireless connection may be used to determine the physical location) [7]. While some existing e-commerce applications are adapted for mobile platforms, the features inherent to mobility and wireless communication medium create a unique class of emerging m-commerce applications striving to achieve the anytime, anywhere paradigm of pervasive computing [11].

In this paper we present our experience with building a prototype of SMMART – **S**ystem for **M**obile **M**arketing: **A**ddaptive, **p**ersonalized and **T**argeted. SMMART is a context-aware application, delivering narrowly targeted promotions to the users wireless mobile devices, such as PDAs and smart phones, when they are in a close proximity or inside a retail store. SMMART adapts to the needs of its user by unobtru-

sively monitoring his/her shopping habits and learning the user's personal preferences. The functionality of the system may be described using a simple metaphor:

*Whenever you go to a retail store, there are brochures advertising current sales. You may be interested in some products, but have no time or intention to look through all pages in the brochure with no guarantee of finding anything interesting. Imagine that there is a genie who knows all about your shopping interests that will carefully read the entire brochure and clip only those promotions that precisely match your interests. SMMART is that genie running on your PDA or smart phone, which can work at any retail store equipped with the corresponding technology. Moreover, it will know when your interests change as long as you continue using it.*

This paper is organized as follows: Section 2 describes related work; Section 3 described a scenario of using SMMART; Section 4 discusses SMMART architecture; Section 5 concludes the paper and presents possible directions of future work.

## 2 Background and Related Work

Varshney and Vetter [12] provide a classification of m-commerce applications, which includes a category of mobile advertising applications that typically use demographic or other information specified by the consumers to deliver targeted advertising messages [14]. These applications may be location-sensitive, delivering the message only to the users that are located in the vicinity of the retailer being advertised [11]. However, coverage area of such applications depends on the precision of the user location determined by the network technology used for wireless connectivity. Each mobile advertising application should cover a small area and narrowly target its recipients to avoid network congestion and overwhelming consumers with a large number of irrelevant advertising messages.

A context-aware system operates and adapts itself based on the knowledge about its user's state and physical surroundings [12]. One of the methods to obtain location context without gathering precise geospatial data is by detecting a connection to a wireless personal area network (PAN), such as WiFi or Bluetooth. Context-aware services enabled by PAN technologies can only reach customers located within a close physical proximity of the wireless service provider. eNcentive framework described in [10] is a context-aware m-commerce application used to distribute electronic coupons. However, it pushes all available coupons to its users regardless of their preferences. To be effective, eNcentive is deployed at a large number of retail sites and requires an even larger number of customers carrying wireless PDAs.

SMMART belongs to the same class of applications as eNcentive and is used to deliver targeted marketing information to customers whose preferences match products that are currently on sale at retail stores. SMMART guarantees a high level of privacy because it does not transmit any personally identifiable information and cannot be used by retailers to track their customers and their buying habits.

SMMART is an example of a user-centric, context-aware pervasive system. In general, pervasive computing systems have the following characteristics [5]: ubiquitous access, context awareness, intelligent behavior, and natural interaction. Ubiquitous access and ubiquitous computing, introduced by Weiser [13], refer to an environment where users are surrounded by computational power and applications. Sen-

sors, smart phones, pagers, PDAs, different miniaturized and embedded devices are the hardware environment supporting ubiquitous services and applications.

Ubiquitous services are context-aware in the terms of location-awareness, time awareness, device-awareness and personalization [6]. Context-awareness refers to the ability of a system to recognize users, to interpret context information and to run in an appropriate fashion for the users, applications and services.

Intelligence in pervasive computing comprises adapting to user behavior, personalization of application and services and supplying users with information at the right place and time. SMMART is designed and implemented as an intelligent pervasive system with abilities to adapt, to target user with information and to personalize its services. SMMART incorporates a number of features enabling it to adapt its behavior based on the current context and past user input.

Natural interaction in pervasive computing refers to system modality where the same functionality is delivered through voice (speech recognition and synthesis), wireless interfaces, and gesture recognition. SMMART is designed to be extensible with a provision to eventually develop a multimodal voice-enabled interface with speech recognition and synthesis.

### 3 Using SMMART

In this section we present a scenario of how a hypothetical shopper named Bob could use SMMART in his everyday shopping (Fig. 1).

Bob recently installed SMMART Client software onto his wireless PDA, entered his musical preferences, as shown in Fig. 2a, and drove to his favorite place to buy CDs. As Bob enters the store, his SMMART Client makes a connection with the store's SMMART Server and tells the server about his preferences. The server responds with a list of products that match Bob's preferences and are currently on sale, as shown in Fig. 2b. Bob selects *Every Breath You Take* on the screen of his PDA to view more information about the promotion. As Bob clicks on this product, his SMMART Client assumes that he may be interested in other products by *Police* and its musicians. In this case, keyword *Sting* (the lead singer of *Police*) is automatically added into the list of Bob's preferences. Bob also decides to purchase *On Every Street* by *Dire Straits*. Clicking on this product description has two consequences: Bob's interest in *Dire Straits* is confirmed and the keyword *Mark Knopfler* (the founder of *Dire Straits*) is added to his preferences.

Later Bob decides to visit a bookstore. At this moment, Bob's preferences include five keywords, as shown in the lower portion of Fig. 1. Upon entering the store, his

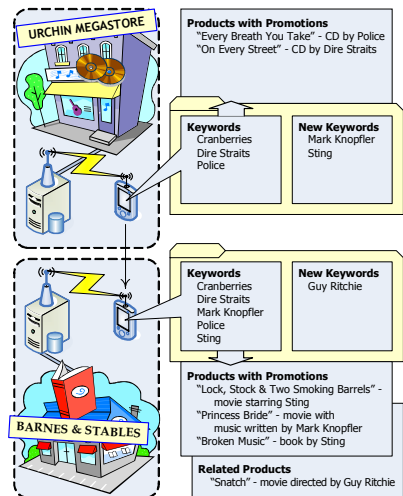


Fig. 1. A typical scenario of using SMMART

SMMART Client receives all current promotions matching his interests, presenting Bob with three products. Bob is most interested in *Lock, Stock and Two Smoking Barrels*, a movie starring *Sting*. As Bob selects this product description, Bob's interest in *Sting* is confirmed and his preferences are updated with a new keyword – *Guy Ritchie* (director of this movie). Bob's SMMART Client also offers a list of related products, which include *Snatch*, a movie directed by *Guy Ritchie*.

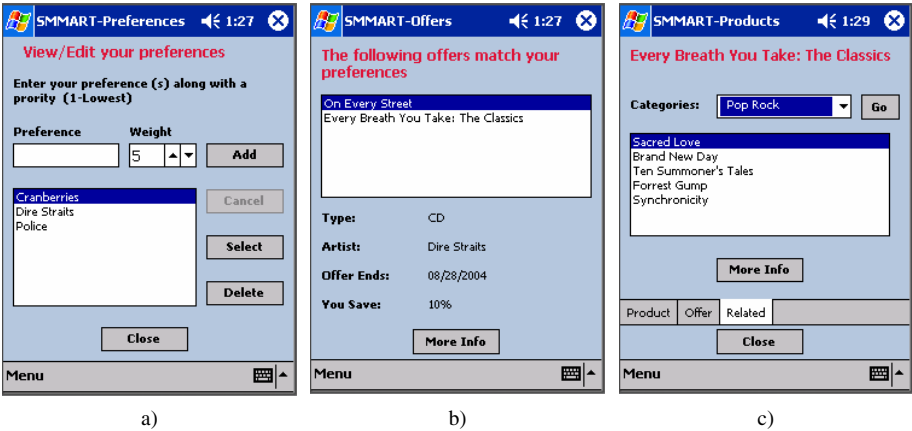


Fig. 2

## 4 Overview of SMMART Architecture

As illustrated by the above scenario, SMMART consists of a server installed at every participating retail location and clients for mobile wireless devices that pull information from the server (Fig. 3). An inventory database of a retail store provides the basis for all data available to the SMMART Server. Product Manager retrieves all relevant information about a specific product, which is then used by SMMART Client when the user chooses to view the details about a particular offering. Similarly, Promotion Manager retrieves all promotion information for a given product. Search & Match Agent is the core of the SMMART Server. This agent receives a list of keywords from the client ordered by their relevancy to the user's interests. For each keyword in the list, the agent finds all matching products that currently have a promotion and adds them to the result. The result consisting of matching products is sorted in the order of relevance to the user preferences and returned to the client.

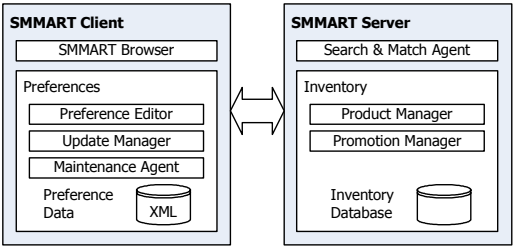


Fig. 3. Architecture of SMMART

As shown in Fig. 3, SMMART Client consists of two main components: SMMART Browser and Preferences module. Typically, after SMMART Client connects to and communicates with a server, the user is presented with a list of matching products that currently have promotions (Fig. 2b). The user explores each product in detail by viewing its full description in SMMART Browser. This information is divided into several sections: product information, offer details and related products (Fig. 2c), which can also be explored using the SMMART Browser.

Preferences module contains and manipulates Preference Data – a list of keywords stored in XML format. Each keyword is associated with a numeric weight representing its relevancy to the user's interests and a timestamp indicating the last time this keyword or its weight was updated. As described below, the keyword weights and timestamps also facilitate the learning capabilities of SMMART.

When the user runs SMMART Client, the Preference Data is scanned. If no keywords are found, the user is prompted to add new keywords using Preference Editor (Fig. 2a). Main responsibilities of this module include adding new, editing and deleting existing keywords, manually changing the weights of existing keywords if needed.

When the user indicates an interest in a particular product by viewing its full description in SMMART Browser, Preference Data is automatically updated by Update Manager. If the keywords associated with a product are already present in Preference Data, their weights are incremented. Otherwise, they are added with a default weight.

SMMART assumes that by viewing full product description the user signifies his interest the product. It is possible for keywords not reflecting true user interests to be automatically added to Preference Data. The primary goal of Maintenance Agent is to detect and remove any keywords that are erroneous or represent past interests of the user. Maintenance Agent runs at application startup and looks for keywords that have not been updated within a specified period and decreases their weight. Eventually, such keywords will be placed in the Recycle Bin and will not be used to find matching products.

## 5 Implementation of SMMART Prototype: Lessons Learned

To prove the functional feasibility of the SMMART, we implemented its fully operational prototype. We chose C# and ASP.NET to implement SMMART Server running on Microsoft .NET Framework. SMMART Client is also implemented in C# running on Microsoft .NET Compact Framework. Our inventory database runs on SQL Server 2000. All tests were conducted using Dell Axim PDAs. The remainder of this section discusses different approaches to the specific implementation details of a context-aware mobile marketing application, such as SMMART, as well as some challenges that we faced in that process.

### 5.1 SMMART Context Information

SMMART uses several types of context information [2] as described below.

**Physical location context** is needed to determine which store's inventory is to be searched every time a user wishes to use SMMART. Our application does not require the knowledge of geographical coordinates of the user's location. Instead, we are using the information about the physical proximity of the user (SMMART client) to

the store (corresponding server). Our approach to obtaining this type of context information is discussed in the next section.

**User context** (user identity) determines what specific information is presented to the shopper. For example, it is reasonable to expect that two different users of SMMART visiting the same store will see a different set of offers because they have different their shopping preferences. In SMMART, user context is also affected by the previous experiences of each user. In the beginning of using SMMART, it is the user's responsibility to enter some keywords describing his or her shopping preferences (if no preferences are entered, the user will have an option to see all offers available at a store). As soon as the user begins browsing through the available offers, SMMART starts analyzing the user's browsing patterns by matching and updating keywords describing each viewed product and keywords in the Preference Data.

## 5.2 Infrastructure of Client-Server Communication

XML Web Services appear to be a good choice for the logical structure of SMMART client-server communication [9]. Firstly, web services fit well in the general philosophy of SMMART: a number of functionally and semantically related methods are united under the umbrella of a single service; all methods work with the same data, i.e. the store's inventory. Secondly, using web services helps overcome the burden of possible network disconnections due to the statelessness of the connection. However, XML and SOAP add a sizeable overhead to the amount of the exchanged data [1], which may result in a delayed application response and congestion of the wireless network connection.

Our primary objective was to prove the viability of SMMART concept. Current implementation of the prototype uses XML web services over a Wi-Fi wireless LAN. Such a choice of technologies works well for a large class of existing Wi-Fi-enabled PDAs. This also assumes that each SMMART site must be equipped with one or more wireless access points. Also, each site must route all network traffic from an access point only to the web server hosting SMMART web services. This enables an unambiguous identification of the store to which a SMMART Client is connected. Additionally, such a routing scheme prevents possible hijacking of the wireless bandwidth. However, there is another class of devices, which includes smart phones equipped with Bluetooth sensors requiring a different combination of network and data access technologies. SMMART can easily be implemented to work with this set of client hardware with no modifications to the architecture of the system.

## 5.3 Design for Handheld Devices

Designing applications for handheld devices is greatly influenced by their hardware limitations, primarily small screen, slow CPU, small amount of RAM, and short battery life. In an application such as SMMART, only the most essential information must be displayed on the screen. This is not only because the screen is small, but also because this application is used in an environment where the user may be easily distracted by many environmental factors. SMMART requires minimal data processing since its algorithms are simple and produce no noticeable delay on Dell Axim PDAs, on which SMMART prototype was tested.

The most challenging issue in designing a networked application for a PDA is the short battery life. Currently, maintaining a WiFi connection on a PDA is a very energy-consuming task. A SMMART client requires wireless connectivity for browsing, searching or matching of any products in the store inventory. A connection is not required for editing of preferences. However, while running a SMMART client, a PDA can be powered off at any moment. When it is turned back on, possibly at a different location, a running SMMART client will detect the changes in the wireless network, find an available SMMART server and obtain a new set of products matching the user's preferences.

#### **5.4 End-User Acceptance**

When a new application arrives on the market, it is crucial to know whether end-users will find it intuitive and easy to use. If the users do not want to use the system, it does not matter how technologically advanced it is or how much savings it could yield. We conducted a survey of potential end-users of SMMART who were given an opportunity to test its prototype using their own shopping preferences in our "test store" containing about one hundred products. The results of our survey indicate that its participants have a very favorable opinion about SMMART. Specifically, based on their own experience with SMMART, 80% of the survey respondents agreed that the system makes good matches between their shopping preferences and products in the test store. Given a chance to browse through the products found as a result of matching of preferences or searching for keywords, navigate through the different screens of the user interface and system options, 80% of the respondents agreed that interface of SMMART Client is intuitive and easy to use. Finally, 93% of the respondents said that if they owned a mobile device running a SMMART Client, they would be willing to use the system in their everyday shopping.

Consumers are always concerned about their privacy: why would they give away potentially compromising information about themselves and their preferences? In terms of preserving the user's privacy, using SMMART is equivalent to searching the inventory of a store with an Internet portal. In this process, the store can deduce the consumer's interest in certain products. A typical online store can also easily detect whether a particular search resulted in a purchase. SMMART enables consumers to make such searches completely anonymous because stores cannot make a connection between a search and a purchase. Additionally, while performing a search, SMMART filters and sorts the obtained results according to the criteria of their relevance to multiple keywords. This effectively eliminates the necessity to reformulate the search query, which arises frequently in searching the inventories of online stores.

#### **5.5 Retailer Acceptance**

Increased revenue is the primary factor that determines the acceptance of SMMART by retailers. Deploying SMMART at a single retail store or at a chain of affiliated stores must be economically justified. The costs of the framework, its supporting infrastructure, data upkeep and maintenance must be less than the revenue from additional sales generated by the customers using SMMART.

At the same time, retailers should not view SMMART as a potential tool to drive up the competition. It is in the retailers' best interests not to allow shoppers to com-

pare products easily, but rather to distinguish their products from the competitors, which can be achieved through personalization. SMMART is designed for use at only one store at a time and therefore shoppers will be unable to compare prices among different stores. This feature should be appealing to the retailers because it creates an easy way to automatically create personalized shopping lists without any investments in additional demographic and market research.

To demonstrate the increase in revenues, we created a simulation model, in which we measured a relative increase in sales generated by purchases resulted from product matches and recommendations made by SMMART. Our experimental results show that SMMART yields the highest increase in sales with the low values of  $P(c)$ , the metric we used in our model, which represents the probability that a customer  $c$  would make a purchase uninfluenced by SMMART. This is typical for upscale stores in shopping malls, stores that sell large ticket items, or stores where people come to socialize, as well as to shop. For example, according to our data, when the probability of a customer to make a purchase is 20% and when only 5% of all customers are carrying SMMART-enabled mobile devices ( $S(c) = 5\%$ ), using the system would yield an almost 13% increase in sales. Alternatively, with higher values of  $P(c)$ , which are typical for stores where customers are determined to make a purchase and stores where customers make routine purchases, such as grocery stores and supermarkets, the expected impact of SMMART is more modest. With  $P(c) = 90\%$  and  $S(c) = 5\%$ , SMMART yields slightly less than a 3% increase in sales.

## 5.6 The Big Picture

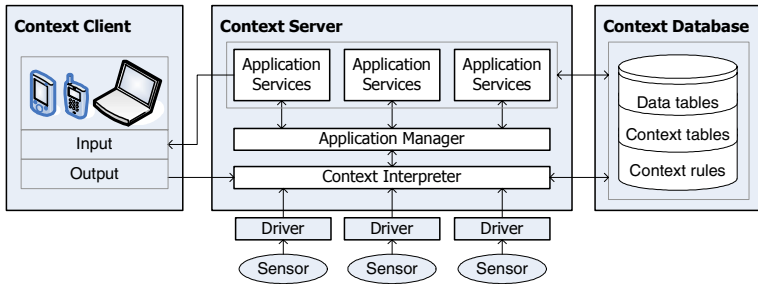
Following our experience with SMMART, we propose a generic client-server architecture for context-aware systems that subsumes a number of other architectures proposed in the literature [3, 8]. Our architecture comprises four core components: sensor information and drivers, context client, context server, and context database.

Different sensors remotely or locally connected to the server, usually network-based, are responsible to supply context information – location, time, device or object status, and personalization. The drivers are software components that interpret sensor information and convert it in appropriate context information for Context Interpreter. Some authors called the drivers widgets [4], or adaptors [6].

Context Interpreter is responsible for converting context information received from the drivers or from the context client input to higher levels of context information understandable by an application and its services. For example, physical coordinates can be converted to street name and number and/or building and floor. If the context can be recognized and interpreted, the context is transferred to the Application Manager, which runs an application or applications connected to the current context. If the context cannot be recognized because of ambiguous, insufficient or inaccurate information, the Context Interpreter queries the context tables and uses context rules in attempt to find the right context. The context tables contain user context information: user preferences, user habits, past user schedules and activities. The context rules are similar to knowledge database rules. Context Interpreter acts as an inference engine to find the right context and to deliver it to the Application Manager.

Application Manager matches context information to applications and services and initiates their execution controlling the running and stopping of services and





**Fig. 4.** Generic Architecture for Context-aware Systems

applications. The applications can query the data tables of the context database to retrieve information and to deliver it to the clients as output.

The services are components that execute actions on behalf of applications (for example turning the lights on or delivering notifications). Usually they are implemented as software agents. The services proactively monitor user calendars and schedules, email accounts, and deliver information to the user in a timely manner.

## 6 Conclusion and Future Work

In this paper we presented a novel approach to mobile marketing in context-aware environments. We built a prototype of SMMART implementing this approach. A user of a wireless PDA equipped with SMMART can receive promotions from retail stores for those products that match his interests. We also studied the economic feasibility of SMMART that indicate that it will be extremely effective in stores where customers need additional incentives to make purchases. Possible examples include stores in shopping malls, bookstores, consumer electronics warehouses, and any other retailers where consumers come not only to shop, but also to socialize.

SMMART can be extended by providing more features that would enhance its usability. A product inventory search would allow a user to search the entire inventory of a store. Product information pages of SMMART Browser could be enhanced with a map schematically showing the location of the selected product in the store. SMMART could also provide the user with the ability to reserve an item which is on back order at the sale price or the option of ordering an in stock item to be picked up and purchased on a designated date and time. This feature could also work well with large items such as big screen TV's or other products where the inventory is not kept on the display floor. Finally, SMMART could be extended with a multimodal interface giving the users an ability of voice communication with the system.

## References

1. H. Chu, C. You, C. Teng. "Challenges: Wireless Web Services," In Proceedings of 10<sup>th</sup> International Conference on Parallel and Distributed Systems (ICPADS'04), July 7-9, 2004, Newport Beach, CA.
2. Dey. "Understanding and Using Context" in Personal and Ubiquitous Computing, Vol. 5, No. 1, pp. 4-7, Springer-Verlag 2001.

3. Dey, G. Abowd. "The Context Toolkit: Aiding the Development of Context-Aware Applications," In Proceedings of the Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6, 2000.
4. Dey, G. Abowd, and D. Sabler. "A Context-Based Infrastructure for Smart Environments," In Proceedings of the 1<sup>st</sup> International Workshop on Managing Interactions in Smart Environments, Dublin, Ireland, Dec. 13-14, 1999.
5. Fersha. "Coordination in Pervasive Computing Environments," In Proceedings of 12<sup>th</sup> IEEE International Workshop on Enabling Technologies, 2003.
6. T. Hofer et al, "Context-Awareness on Mobile Devices – the Hydrogen Approach," Proceedings of 36<sup>th</sup> Hawaii International Conference on System Sciences, 6-9 January, 2003.
7. P. Kannan, A. Chang, A. Whinston. "Wireless Commerce: Marketing Issues and Possibilities," In Proceedings of The 34<sup>th</sup> Hawaii International Conference on System Sciences, IEEE CS Press, 2001.
8. S. Meyer, A. Rakotonirainy. "A Survey of Research on Context-Aware Homes," In Proceedings of the Workshop on Wearable, Invisible, Context-Aware, Pervasive and Ubiquitous Computing, Adelaide, Australia, 2003.
9. T. Pilioura, T. Tsalgatidou, S. Hadjiefthymiades. "Scenarios of Using Web Services in M-Commerce." ACM SIGecom Exchanges. Vol. 3, No. 4, Jan. 2003, pp. 28-36.
10. O. Ratsimor, T. Finin, A. Joshi, Y. Yesha. "eNcentive: A Framework for Intelligent Marketing in Mobile Peer-to-Peer Environments," In Proceedings of The 5<sup>th</sup> International Conference on Electronic Commerce (ICEC-03), Pittsburg, PA, October 2003.
11. U. Varshney. "Location Management for Mobile Commerce: Applications in Wireless Internet Environment." ACM Transactions on Internet Technology. Vol. 3, No. 3, Aug. 2003, pp. 236-255.
12. U. Varshney, R. Vetter. "Mobile Commerce: Framework, Applications and Networking Support." Mobile Networks and Applications, Vol 7, pp. 185-198, Kluwer, 2002.
13. M. Weiser. "The Computers for the 21st Century", Scientific American, 265, 3, September 1991.
14. H. Yunos, J. Gao, S. Shim. "Wireless Advertising's Challenges and Opportunities." IEEE Computer, Vol. 36, No. 5, pp. 30-37, IEEE CS Press, 2003.