

Emerging Ubiquitous Knowledge Services: From Mobile Sensing to Ubiquitous Crowdsourcing and Beyond

Uichin Lee, Howon Lee, Bang Chul Jung, Junehwa Song

Dept. of Knowledge Service Engineering, Korea Advanced Institute of Science and Technology

Institute for Information Technology Convergence, Korea Advanced Institute of Science and Technology

Dept. of Information and Communication Engineering, Gyeongsang National University

Dept. of Computer Science, Korea Advanced Institute of Science and Technology

ucllee@kaist.ac.kr, hwlee@itc.kaist.ac.kr, bcjung@gnu.ac.kr, junesong@kaist.ac.kr

Abstract:

Recent advances of information and communication technologies (e.g., smartphones and Web 2.0) have facilitated ubiquitous sensing and networked collaboration among people. This trend has significantly lowered the knowledge transfer barriers and has broadened the range of knowledge services, thus enabling new types of services called “ubiquitous knowledge services.” The goal of ubiquitous knowledge services is to seamlessly integrate content (e.g., data, information, knowledge) from various sources (e.g., content providers, sensors, smart devices, humans) at large scales and to derive new values for end users in ways that the contributor of the content even did not plan or imagine. In this chapter, we review the key enabling technologies of ubiquitous knowledge services and the existing integrated platforms on sensor data collection, retrieval, and application services. Also, we propose a set of desirable features of the ubiquitous knowledge service platforms based on our vision of the knowledge communications service (KCS).

1 Introduction

While traditional knowledge services are mainly delivered by the experts with domain specific knowledge, recent advances of information and communication technologies (e.g., smartphones and Web 2.0) have dramatically changed the nature of knowledge services. These technologies have facilitated ubiquitous sensing and networked collaboration among people, thus significantly broadening the range of knowledge services and lowering the knowledge transfer barriers. People generate significant amount of content and share it with other users via Web 2.0 services (e.g., Youtube, Twitter). Collaboration over the Internet is common as one can easily create an online project with tens of thousands of participants (called crowdsourcing), e.g., Wikipedia, online Q&A services, and Innocentives. Further, sensing happens everywhere, ranging from surveillance cameras on the streets to smartphone sensors (e.g., collecting mobility traces of users) to even human sensors (e.g., sensing social events). This dramatic shift of ubiquitous sensing and collaboration has enabled new types of knowledge services called “ubiquitous knowledge services.” The key role of ubiquitous knowledge services is to seamlessly integrate content from various sources at large scales and to derive new values for end users in ways that the contributor of the content did not plan or imagine.

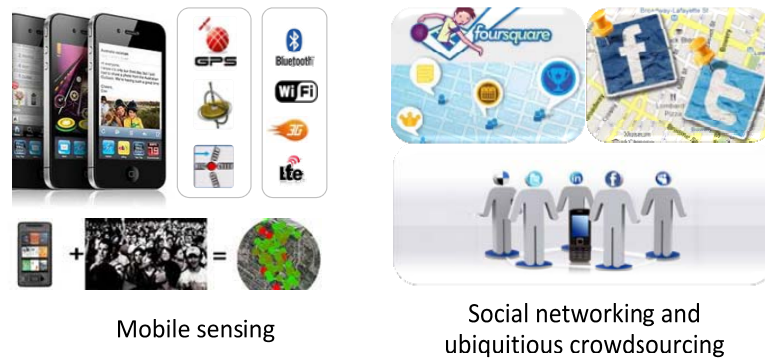


Figure 1 Mobile sensing, social networking, and ubiquitous crowdsourcing

To illustrate ubiquitous knowledge services, consider the case of a recent flooding happened in July 2011 in Seoul, South Korea, which is reported as the heaviest rain falls ever recorded. Due to heavy rain falls in a relatively short period of time, many places were flooded, and the city government was not able to properly track them. Yet people took photos/videos of the local areas and shared them with others via social networking services (e.g., Twitter, Facebook). Given this, various knowledge services were instantly created to help people to make informed decisions; e.g., one service collected all the news feeds from social networking and visualized related photos/videos on the map. Another case is delivering traffic information to the drivers. Traditional traffic information systems mostly use expensive sensors for traffic condition estimation (e.g., fixed cameras or induction loops on the roads), and yet, the coverage of those systems is extremely limited due to high installation and maintenance costs. An alternative is to let drivers to use their smartphone apps to gather/share the traffic information (Thiagarajan, et al., 2009). While individual reports may not be as accurate as traditional approaches, this crowdsourced knowledge service can *collectively* deliver accurate information even with extended coverage and minimal infrastructure costs. Other knowledge services include cooperatively estimating/sharing the schedule of public transits (Thiagarajan, Biagioni, Gerlich, and Eriksson, 2010) or traffic signals (Koukoumidis, Peh, Martonosi, 2011). These use cases (and many to come) clearly demonstrate that ubiquitous sensing and collaboration provide unique opportunities for enabling various ubiquitous knowledge services.

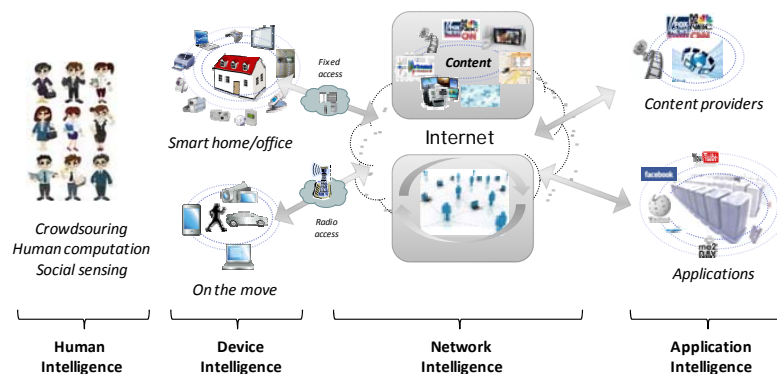


Figure 2 Driving forces of emerging ubiquitous knowledge services

As illustrated in Figure 2, we claim that the main driving forces of emerging ubiquitous knowledge services are the intelligences on device/network (e.g., smart devices, mobile sensing, and in-network processing), application (e.g., Web, service mashup and cloud computing), and human (e.g., crowdsourcing, human computation, social sensing).

- Recent smart devices such as smartphones and vehicles are equipped with various sensors (e.g., GPS, cameras, accelerometers) and wireless networking technologies (e.g., Bluetooth, Wi-Fi, and 3G). Sensing capabilities with always-on connectivity enabled a variety of new mobile sensing applications such as

people sensing and vehicular sensing. Further, the network infrastructure is becoming more intelligent as it supports various services ranging from simple quality of service control to in-network data processing and application services (e.g., machine-to-machine and Internet of Things).

- Application intelligence has facilitated a rapid innovation of application service developments. For instance, service mashup techniques allowed application developers to easily use and combine data, presentation or functionality from multiple sources to create new services. Cloud computing enabled ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- The areas of human intelligence include the capability of collaboration among people to achieve a certain goal (e.g., crowdsourcing) and the capability of human computation where humans perform *the computation tasks* that machines cannot do well (e.g., image labeling, natural language processing) or *the perception tasks* by sensing/processing ambient/social environments (e.g., earthquake reports, social events).

In this chapter, we begin by providing a detailed review of these areas to better understand ubiquitous knowledge services with emphasis on device, network and human intelligences. In particular, we review various mobile sensing and ubiquitous crowdsourcing applications for ubiquitous knowledge services, namely monitoring personal environmental impact (e.g., CO₂ generation/exposure), analyzing context-awareness with sound sensing, sharing personal context over social networking, monitoring road surface (e.g., potholes) and real-time traffic, facilitating fuel-saving driving, and social sensing with Twitter (e.g., weather monitoring and earthquake detection).

Given that the fundamental goal of ubiquitous knowledge services is integrating content (e.g., data, information, knowledge) from various sources (e.g., sensors, databases, humans) at large scales and driving new values to end users, we review the existing systems on sensor data collection, retrieval, service systems, ranging from research prototypes to recent standards such as machine-to-machine communications, EPCglobal Network, and Semantic Sensor Web. Since our ultimate goal is to design an open platform for ubiquitous knowledge services, we also suggest the desirable features based on our insights from the review, namely semantic query processing, service-device/network interaction (for cross-layer optimization), human computation/sensing, social networking, and agent support. Beyond simple data collection and processing, ubiquitous knowledge services naturally require complex semantic query processing at large scale. Service-device/network interactions are necessary as we need to optimize query processing over the network of resource constrained smart devices. Social networking and collaboration will enable various crowdsourcing and human computation/sensing projects. We also anticipate that future knowledge services will be more proactive to achieve better user experiences. For example, user agents perceive the intention of a user and provide the customized knowledge services to the user at any place and any time, with any device even with physical restrictions. Having said that we finally present our vision platform called the knowledge communications service (KCS) towards the goal of realizing these features.

2 Fundamentals of Smart Devices

A smart device is generally referred to an active computer networked device with intelligence (see Figure 3). Mark Weiser proposed three basic forms for ubiquitous system devices, namely tabs, pads and boards (Weiser, 1999). Tabs are accompanied or wearable centimeter sized devices, e.g., smartphones, pads are hand-held decimeter sized devices, e.g., laptops and iPads, and boards are meter sized interactive display devices, e.g., horizontal surface computers and vertical smart boards. These smart devices could be embedded into home appliances (e.g., smart TVs and refrigerators) and vehicles (e.g., SatNav and vehicular sensors). Also, dedicated sensors for monitoring physical environments as in sensor networks can be considered as smart devices. Note that recent smart devices are equipped with various sensors. Typical sensors include motion sensors such as accelerometer, magnetometer (digital compass), and gyroscope (rotational motion and changes in orientation), an ambient light sensor (brightness of the light), a proximity sensor (presence of nearby objects), a camera/video, a voice sensor, and location sensors. In this section, we review the capabilities of smart devices, e.g., processing, wireless networking, localization, multimodal interfaces.



Figure 3 Examples of smart devices

2.2 Processing Capabilities of Smart Devices

We review the system configuration of smart devices, namely memory, CPU, and storage capacity. We show that unlike regular computing systems such as PC and laptops, smart devices are resource constrained, and thus, the resource constraints must be carefully considered when designing knowledge services.

Most smart devices such as use flash memory to store operating systems and data (Khatib, van der Zwaag, Hartel and Smit, 2007). They typically use the shadowing technique such that during system booting time, the entire code image of an OS and applications is copied from flash memory to DRAM for execution. Low-end smart devices such as SatNavs are typically equipped with 64-128MB DRAM and 1-2GB of flash memory; high-end smart devices such as smartphones and smartpads are equipped with 512-1GB DRAM and 16-32GB of flash memory. For instance, Clarion EZD580 SatNav has 64MB DRAM and 2GB NAND flash memory; TomTom GO910 has 64MB DRAM and 1GB flash memory. iPhone 4 and Samsung Galaxy S smartphones are equipped with 512MB DRAM and 16-32GB flash memory; smartpads such as iPad 2 and Galaxy Tab have higher memory of 1GB. Smart devices tend to keep the DRAM size minimal because DRAM is power hungry; each DRAM refresh cycle dissipates a few milliwatts per MB (Khatib, van der Zwaag, Hartel, and Smit, 2007). Moreover, it is known that for a given workload, there are some threshold values for DRAM and flash memory sizes such that increasing the size beyond those threshold values will not bring any further performance gain (Douglass, et al., 1994). For instance, smartphones have about 512MB DRAM. The trend is also true in recent multimedia systems. For instance, Apple TV has a 1 GHz Pentium M Crofton Processor, 256MB DRAM, and a 40G HDD. Smart devices typically use low-power and -cost CPUs such as ARM and xScale whose clock speed is around 1Ghz, or low-power/cost multimedia processors such as Texas Instruments OMAP 2/3 and RMI Alchemy Au series. Although Intel recently released a high performance mobile CPU called Atom (Silverthorne 1.6/8Ghz), its performance is about the same as 900Mhz Celeron M.

In summary, we observe that smart devices are limited in terms of DRAM and CPU power, by a few orders of magnitude, compared to standard desktop machines or servers. The recent trend suggests that such a computing/memory resource gap will continue to exist between smart devices and regular desktop machines in the foreseeable future.

2.2 Wireless Networking Technologies for Smart Devices

Popular wireless networking technologies for smart devices include cellular networks (3G/4G), wireless local area networks (WLANs), personal/body area networks (PANs and BANs), and vehicular communications (DSRC/WAVE). The cellular networks are for long range communications, whereas wireless LANs and vehicular communications (DSRC/WAVE) are for medium and short range communications. Personal/body area networks such as Bluetooth and ZigBee are used for only short range communications. In Figure 4, we present the wireless networking standards in terms of range and data rate.

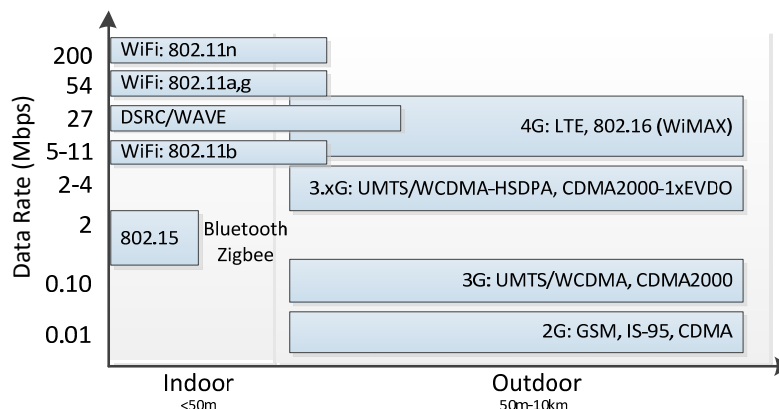


Figure 4 Wireless networking standards (range vs. rate)

Cellular Networks: Cellular systems have been evolving rapidly to support the ever increasing demands of mobile networking. 2G systems such as IS-95 and GSM support data communications at the maximum rate of 9.6kbps. To provide higher rate data communications, GSM-based systems use GPRS (<171kbps) and EDGE (<384kbps), and IS-95-based CDMA systems use 1xRTT (<141kbps). Now 3G systems support much higher data rate. UMTS/HSDPA provides maximum rates of 144kbps, 384kbps, and 2Mbps under high mobility, low mobility, and stationary environments respectively. CDMA2000 1xEvDO (Rev. A) provides 3Mbps and 1.8Mbps for down and up links respectively. 4G systems such as 3GPP LTE and IEEE WiMAX provide the data rate (tens of Mbps), mobility (<60Km/h), and coverage (<10Km) required to deliver the Internet access to mobile clients, which are in the process of deployment.

WLAN: Wi-Fi supports broadband wireless services. 802.11a/g/n provides up to several hundreds of Mbps and has nominal transmission range of 38m (indoor) and 140m (outdoor). Despite its short radio range, its ubiquitous deployment makes Wi-Fi an attractive method to support broadband wireless services. Wi-Fi supports direct peer-to-peer communications called Wi-Fi Direct, a standard that allows Wi-Fi devices to talk to each other without the need for wireless access points (hot spots). Wi-Fi Direct enabled devices can signal other devices to make connections and can also form a Wi-Fi Direct Group for device to device communications.

Bluetooth and ZigBee: Bluetooth is an always-on, low-power, and short-range hookup for implementing Wireless Personal Area Networks (WPANs). It became the most popular PAN communication device. The concept behind Bluetooth is to support universal short-range wireless capabilities using the 2.4GHz unlicensed low power band. Bluetooth allows users to connect up to eight devices by forming a star-shaped cluster, called piconet. The cluster head is called master and the other nodes are called slaves. Two Bluetooth devices within 10m range can exchange data up to 2Mbps (Bluetooth v2.0). Readers can find more information about Bluetooth and its applications such as content distribution in the work by Lee, et al. (2010). Another WPAN radio is ZigBee, a low-cost, low-power, wireless network standard with data transmission rates of 20-250 kbps. The low cost allows the technology to be widely deployed in wireless control and monitoring applications. Low power-usage allows longer life with smaller batteries. ZigBee supports P2P style mesh networking that is simpler and less expensive than Bluetooth.

Vehicular Communications: Dedicated Short-Range Communication (DSRC) is a short to medium range communication technology operating in the 5.9 GHz range (Jiang, et al., 2006). The Standards Committee E17.51 endorses a variation of the IEEE 802.11a MAC for the DSRC link. DSRC supports vehicle speed up to 120mph, nominal transmission range of 300m (up to 1000m), and default data rate of 6Mbps (up to 27Mbps). This will enable operations related to the improvement of traffic flow, highway safety, and other Intelligent Transport System (ITS) applications in a variety of application environments called DSRC/WAVE (Wireless Access in a Vehicular Environment). DSRC has two modes of operations: (1) Ad hoc mode characterized by distributed multi-hop networking (vehicle-vehicle), (2) Infrastructure mode characterized by a centralized mobile single hop network (vehicle-gateway). Note that depending on the deployment scenarios, gateways can be connected to one another or to the Internet, and they can be equipped with computing and storage devices, e.g., Infostations (Frenkiel, Badrinath, Borrás, and Yates, 2002).

2.3 Localization

We review localization techniques that are commonly used by the smart devices, namely GPS, A-GPS, network positioning systems (or cell-tower localization), and Wi-Fi Positioning Systems (WPSs).

The Global Positioning System (GPS) is a space-based global navigation satellite system that provides location and time information. There are 27 satellites in the orbit each of which carries highly accurate atomic clocks and is controlled by the ground control station. The orbits are arranged so that at anytime, anywhere on Earth such that there are at least four satellites "visible" in the sky. A GPS receiver's job is to locate three or more of these satellites, figure out the distance to each, and use this information to deduce its own location. This operation is based on a mathematical principle called trilateration, a process of determining absolute locations of points by measurement of distances, using the geometry of circles or spheres. Imagine you are somewhere in Korea and you are totally lost—for some reason, you have absolutely no clue where you are. You find a friendly local and ask, "Where am I?" He says, "You are 70 km from Daejeon." You ask somebody else where you are, and she says, "You are 60 km from Daegoo." Now you have two circles that intersect. You now know that you must be at one of these two intersection points. If a third person tells you that you are 100 km from Gwangjoo, you can eliminate one of the possibilities. You now know exactly where you are.

Smart devices that use cellular networks mostly have an Assisted-GPS (A-GPS) that significantly reduces satellite search space by focusing on where the signal is expected to be. A cellular network service provider has an assist server and position server with GPS that collects satellite information (e.g., time sync, frequency, visible satellites, etc.), which is transmitted over cellular network to speed up location fixing. In practice, however, even with such enhancement, the performance of A-GPS in smart devices is less satisfactory. Due to small form factors, smart devices mostly use micro-strip or wire based antenna whose performance is inferior to that of spiral helix antenna that is commonly used in SatNavs.

Another commonly used localization method is cell-tower localization. The overall localization method is very similar to GPS in that a mobile node measures the distances from nearby cell towers (whose positions are known) and trilateration is performed to determine its location. Distance measurement can be done with time-of-arrival of signals or signal-strength measurement. Unlike GPS, such measurements in urban environments are error-prone and thus, the error range is much greater than GPS (hundreds of meters on average).

A Wi-Fi positioning system (WPS) is based on fingerprinting (e.g., RADAR and Skyhook). WPS uses a fingerprinting technique and is generally composed of two steps: training and positioning phases. In the training phase, the system builds a fingerprint table: for each location, the system collects signal strength samples from towers, and keeps the average for each location. In the positioning phase, the system calculates the distance in signal strength space between the measured signal strength and the fingerprint DB; e.g., by selecting k fingers with the smallest distance, and using arithmetic average as an estimated position. One of the major shortcomings is that building the fingerprint table (e.g., wardriving) is costly and the table must be periodically updated as Wi-Fi hotspots tend to change over time.

2.4 Multimodal Interfaces

Basic user interfaces in smart devices are the standard graphic based WIMP (window, icon, menu, and pointing device). Smart devices also have multiple modalities and are able to interpret information from various sensory and communication channels. For instance, smartphones uses touch based interactions; LG Optimus smartphones allow users to use personalized gestures as shortcuts to commands. Oviatt (2003) defined that multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring in the forms of human language and behavior, and which incorporate one or more recognition-based technologies, e.g., speech, pen, vision. Thus, they can provide richer and more natural methods of interactions, e.g., gesture, speech, and even all the five senses.

Multimodal interfaces were initially considered as more efficient than existing unimodal interfaces, but experimental results showed that the speed up of task completion is mere 10%, suggesting that speed-up should not be considered as a primarily measure (Oviatt, 1997). Instead, multimodal interfaces have been shown to significantly reduce the

error rate and improve the reliability; for instance, Oviatt (1997) showed that users made 36% fewer errors with a multimodal interface than with a unimodal interface. People generally prefer to use multimodal interaction due to its expressive power.

3 Application Intelligence

Application intelligence facilitated a rapid innovation of these application services. Key enabling technologies of the application intelligence include service mashup, cloud computing and knowledge visualization. There are several other techniques that contribute to the application intelligence such as context awareness, augmented reality, and privacy/security. The application intelligence permits rapid, efficient software development for the developers, and rich user experience (UX) for the users. In this section, we review the key enabling technologies, namely service mesh-up, cloud computing and knowledge visualization.

In general, service mashup means a web page or application that combines data, service, presentation or functionality from multiple sources to create new services. To make existing data more valuable for personal and professional purposes is very important to provide a better user experience, and this is one of the key success factors in the application and content business. In other words, the mashup is a way to create new web applications by combining existing web resources utilizing data and web APIs. Web services are emerging as a major technology for deploying automated interactions between distributed and heterogeneous applications. Even though many service providers recognized its value and have already adopted the service mashup concept, fully realizing the service mashup concept is still challenging, and much work remains to be done. Since services generally face with a broader range of heterogeneity compared to data, and thus, resolving heterogeneity is the key challenge in the area of service mashup.

Cloud computing delivers hosted services over the Internet. It has enabled ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (Mell and Grance, 2011). Cloud computing describes a new eco-system model for IT services based on Internet protocols, and it typically involves provisioning of scalable and virtualized resources, such as storages and applications.

Knowledge visualization aims to improve the creation and transfer of knowledge among people via various computer or non-computer based visualization methods. While information visualization focuses on the use of computer-supported tools to derive new insights, knowledge visualization concentrates on transferring insights, experiences, attitudes, values, expectations, perspectives, opinions, and predictions with various complementary visualization methods (Burkhard, 2004) (Burkhard and Meier, 2004) (Eppler and Burkhard, 2005). For an effective transfer and the creation of knowledge via visualizations, following four perspectives should be taken into account, such as function type, knowledge type, recipient type, and visualization type. A function type perspective and knowledge type perspective answer why visualization should be used (aim) and what type of knowledge needs to be visualized (content), respectively. A recipient type perspective points to the different backgrounds of the audience, and the visualization type perspective organizes the main visualization types in accordance with their individual properties.

4 Ubiquitous Sensing for Knowledge Services

We present ubiquitous sensing applications for smart devices; for people sensing, we review knowledge services on personal environmental impact monitoring, sound sensing, state sharing over social networking, and for vehicular sensing, we review knowledge services on road surface monitoring, real-time traffic monitoring, and fuel-efficient driving.

4.1 People Sensing based Knowledge Services

4.1.1 Personal Environmental Impact Report (PEIR)

Participatory sensing is to realize distributed data collection and analysis at personal, urban, and global scale. Mobile phones are used to gather data, and web services are used to aggregate and interpret the collected data. Personal

Environmental Impact Report (PEIR) is a mobile personal sensing platform that uses mobile handsets to collect/upload data, and generates web-based reports about personal environment impacts (Mun, et al., 2009). The PEIR system estimates the metrics related to health and wellness: (1) carbon impact: a measure of transportation-related carbon footprint, (2) sensitive site impact: a user's transportation related airborne particulate matter emissions near sites with populations sensitive to emissions, (3) smog exposure: a user's transportation-related exposure to particulate matter emissions, and (4) fast food exposure: the time integral of proximity to fast-food restaurants. The ultimate goal of PEIR is for people to take notice of the impact and exposure and to raise individual/community awareness on the potential risks.

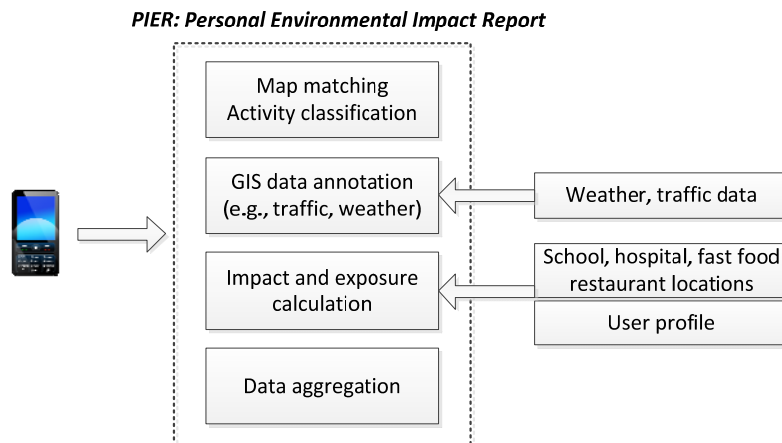


Figure 5 PEIR system architecture

Figure 5 shows how the PEIR system works. PEIR basically collects the location traces from individuals using the onboard GPS in mobile phones. Once the data are uploaded, raw GPS readings are matched with the map to reduce the GPS errors. Then, a user's activity is classified by processing the GPS readings (e.g., walking and driving). PEIR uses external data sources. For each site visited, it annotates the data with weather and traffic information by utilizing external information sources (e.g., government organizations for weather and transportation). The system utilizes the user profile (e.g., vehicle type) and existing location information of schools, hospitals, and fast food restaurants. For a given trace, the carbon and sensitive site impacts are estimated using the automobile emission factor model (EMFAC) customized with a user's vehicle type and weather condition. Similarly, the smog exposure can be estimated by incorporating the traffic information. The fast food exposure can be simply calculated based on the location traces and fast-food locations.

4.1.2 Sound Sensing for People-Centric Applications

Sound captured by the microphone of a mobile phone is a rich source of information that help make inference about the person carrying the phone, the ambient environments, and even the social events. SoundSense is a sound processing application that automatically classifies the sound into several categories of interest (e.g., voice and music) (Lu, et al., 2009). SoundSense is also capable of learning ambient sounds considered significant (e.g., vehicle starting sound, vacuum cleaner sound, etc.). To this end, it incrementally builds probabilistic models of encountered sounds. The detected sounds are then ranked based on importance and are given to the user to perform human-in-the-loop labeling. This ambient sound logs can be used as a diary or can be used to collect/share the sound information across the city to understand the noise level of an urban environment.

4.1.3 Sensing and Social Networking

Given that mobile phones are equipped with sensors such as GPS and accelerometer, they can be used to create mobile sensor networks to sense information important to people. One example would be using mobile phones to automatically detect/publish the current locations or activities of users. CenceMe is a mobile platform that tracks users (e.g., location and activity) and shares the user information over social networks (Miluzzo, et al., 2008). CenceMe's ClickStatus application displays the status of CenceMe buddies from Facebook (see Figure 6). For

privacy management, CenceMe provides a privacy setting GUI through which users can enable/disable sensors used for tracking.

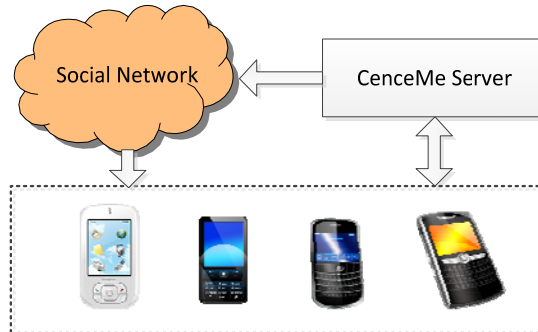


Figure 6 CenceMe overview

CenceMe uses various smartphone sensors to extract user information. Simple detections are performed in the phones; e.g., classifying whether the incoming sound is human voice or environmental noise, and classifying acceleration data to activities (e.g., sitting, standing, walking, and running). For sophisticated analysis (e.g., conversation, social context, and mobility mode), the relevant sensor data are transmitted to the backend servers in which the classification operations are performed.

4.2 Vehicular Sensing based Knowledge Services

4.2.1 Road Surface Monitoring

Municipalities around the world spend millions of dollars to keep roads in good ride quality. Ride quality is mainly measured by the pavement roughness of a road surface that is an expression of the surface irregularity. The roughness causes vertical vibration of vehicles and thus reducing ride quality. There are several main sources of roughness: steps, dips, humps, bumps, and defects (crack, potholes). It also causes the stresses and fatigue damages of both vehicle and road structures. Municipalities have been profiling roads using non-contact profiling devices mounted on the vehicles that use GPS, accelerometer/laser sensors. This approach is quite expensive due to the operation costs and limited availability of specialized profiling devices.

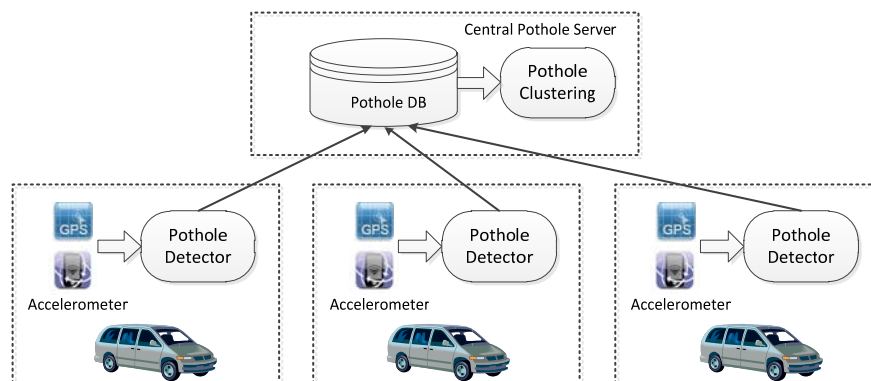


Figure 7 Pothole patrol system overview

Researchers have recently considered using less expensive commodity sensors; e.g., smartphones are good sensing platforms as demonstrated in by Mohan, Padmanabhan, and Ramjee (2008). Eriksson, et al. (2008) developed a pothole detection algorithm that uses off-the-shelf GPS and 3-axis accelerometer sensors. For pothole detection, the acceleration data needs to be carefully examined because potholes must be differentiated from other road anomalies (e.g., railroad crossings and expansion joints). The detection algorithm mainly uses the X-Z ratio since potholes influence the acceleration values only one side of the vehicles (as opposed to both sides in railroad crossings and

expansion joints). It also removes anomalies caused by turning, veering, braking, and also samples collected at high speed. As shown in Figure 7, a mobile node runs the detection algorithm and returns results to the central server. For robust detection, the server clusters at least k events in the same location. The robustness can be further improved when multiple vehicles participate in the monitoring project.

4.2.2 Traffic Monitoring Systems

Most traffic information systems currently analyze data from closed-circuit cameras and sensors installed on the roads to estimate traffic conditions. This information is then shared via radio broadcasts, traffic report on the Internet (e.g., Google Traffic), and en route traffic displays and signals. The coverage of those systems, however, is extremely limited (e.g., mainly highways) due to high installation and maintenance costs. To overcome such limitations, researchers have proposed to use vehicle as sensors to collect GPS measurements. This mobile sensor approach greatly extends coverage, thus enabling *street-level traffic flow estimation*. Yoon, Noble, and Liu (2007) systematically showed that various features can be used for accurate traffic flow estimation, e.g., average speed, speed distribution, speed changes, and time duration/distribution.

For data collection and sharing, we can consider various wireless connectivity such as roadside WiFi Access Points (APs), vehicle-to-vehicle communications, and cellular communications (2/3G). CarTel (Hull, et al., 2006) uses roadside WiFi APs, and TrafficView (Dikaiakos, Iqbal, Nadeem, and Ifode, 2005) and MobEyes (Lee, et al., 2006) use car-to-car communications. Also, cellular communications have been recently considered: TruTraffic by Dash Express's SatNav system and smartphone-based system like VTrack. In particular, TruTraffic collects GPS measurement data from SatNav users using 2/3G and provide a real-time traffic information service such as dynamic route guidance and congestion notification. If smartphones are used, energy consumption and GPS errors must be considered as in VTrack (Thiagarajan, et al., 2009). Adaptive sampling is preferable as GPS is energy hungry. Due to inferior antenna and urban canyon, GPS is not always accurate and even may be unavailable. In this case, alternative location services must be employed (e.g., Wi-Fi and cellular localization). VTrack solves these challenges by employing a novel map matching algorithm that greatly reduces the required number of samples and yet significantly lowers location tracking errors.

4.2.3 Fuel Efficient Driving

Fuel efficient driving can be realized using participatory sensing. We introduce two participatory sensing systems, namely GreenGPS (Ganti, et al., 2010) and SignalGuru (Koukoumidis, Peh, Martonosi, 2011). GreenGPS recommends the most fuel-efficient route to the drivers. SignalGuru predicts the traffic signal schedule as in a Green Light Optimal Speed Advisory (GLOSA) system; when heading towards a signalized intersection, drivers can find the optimal speed that minimizes the waiting time at the intersection (thus, saving fuel).

GreenGPS uses grassroots collaboration among users who have onboard OBD-II sensors that are capable of retrieving detailed vehicle statistics in real-time (e.g., RPM, fuel consumption, speed, etc.). The fuel consumption depends on multiple parameters such as year, model, and make of a vehicle. However, when the number of participants is small, it is challenging to accurately model fuel consumption particularly (requiring a large number of samples). Instead, GreenGPS builds a generalized prediction model with model clustering, i.e., grouping similar items, say the same make and model. The resulting model takes the input parameters such as stop sign count, traffic light count, distance travelled, average speed, and vehicle weight. The experimental results show that an average 10% fuel saving can be achieved with GreenGPS.

SignalGru also relies on information sharing among drivers. Unlike traditional GLOSA systems where dedicated devices are installed at the intersection (e.g., counting down unit or digital sand watch), SignalGru is infrastructure-less as it uses commodity cameras in smartphones to detect signal changes. The detection module periodically processes the video frames to detect signal changes (e.g., Green to Yellow to Red). The detected information is then disseminated via vehicle-to-vehicle communications. Once enough information is collected, the schedule prediction can be performed. Predicting the schedule of pre-timed traffic signals (with fixed schedules) is straight-forward. SignalGru also supports the schedule prediction of traffic-adaptive signals where the schedule dynamically changes on the basis on the current vehicle density (sensed via loop detectors on the roads). The experimental results show that SignalGru can accurately predict the schedule (less than several seconds of an error), and an average of 20.3% fuel saving can be achieved.

4.2.4 Urban Surveillance

Vehicular sensing can be used for proactive urban monitoring services (Lee, et al., 2006) where vehicles continuously sense events from urban streets, maintain sensed data in their local storage, autonomously process them, e.g., recognizing license plates, and possibly route messages to vehicles in their vicinity to achieve a common goal, e.g., to permit police agents to track the movements of specified cars. Vehicular sensing could be an excellent complement to the deployment of fixed cameras/sensors. This completely distributed and opportunistic cooperation among sensor-equipped vehicles has the “deterrent” effect of making it harder for potential attackers to disable surveillance. Another less sensational but relevant example is the need to track the movements of a car, used for a bank robbery, in order to identify thieves, say. It is highly probable that some vehicles have spotted the unusual behavior of thieves' car in the hours before the robbery, and might be able to identify the threat by “opportunistic” correlation of their data with other vehicles in the neighborhood. It would be much more difficult for the police to extract that information from the massive number of multimedia streams recorded by fixed cameras.

5 Ubiquitous Crowdsourcing for Knowledge Services

The term crowdsourcing was first coined by Jeff Howe in a Wired Magazine article (derived from outsourcing). Crowdsourcing is the act of taking a job traditionally performed by an employee and outsourcing it to the public, hoping to exploit the spare processing power of millions of human brains in the distributed labor network over the Internet (i.e., any people behind their computers who are willing to contribute to the project). Example crowdsourcing projects include Wikipedia, social Q&A (e.g., Yahoo! Answer), and Innocentives.¹ The required human skills of crowdsourcing range from simple (e.g., image labeling and transcribing) to complex (e.g., research and development). Note that simple tasks that human perform better than machines (e.g., image labeling and transcribing) are often called “human computation tasks.” The motivation behind crowdsourcing participation largely depends on the types of systems; e.g., monetary rewards in Amazon’s M-Turk², a mechanized labor marketplace, altruism in Wikipedia, and fun in human computation games. Readers can find more detailed information about human computation in the work by Quinn and Bederson (2011).

In recent years, the availability of networked labor forces has significantly improved with smart devices due to always-on wireless connectivity. Further, a variety of sensors in smart devices (e.g., cameras, accelerometers) enable a new type of crowdsourcing, namely sensing tasks (e.g., taking photos from streets) which we call ubiquitous crowdsourcing. Sensing activities can be largely classified into two categories: (1) passive sensing task: data are automatically collected on the background without any user intervention, and (2) active sensing task: a user needs to perform physical activity for sensing (e.g., taking photos). Participatory sensing that we reviewed in Section 3 belongs to this category. An alternative to using sensors is to use a human’s five senses and cognitive ability (inference); e.g., a flooding/earthquake can be easily detected. Soliciting participation in ubiquitous crowdsourcing can be done in a number of ways; e.g., people with the same interest or from the same community as in participatory sensing, mechanized labor forces as in Amazon’s M-Turk, and people from the social networks as in Twitter. In this section, we illustrate several ubiquitous crowdsourcing applications by reviewing existing systems that utilize mechanized labor forces and social networks.

5.1 Location-based Ubiquitous Crowdsourcing

Location-based tasks (e.g., taking photos, price/inventory checks, and verifying map information) are one of the key services of ubiquitous crowdsourcing, and several on-line services include mCrowd³, Field Agent⁴, and Gigwalk⁵. Microsoft Bing has been collecting photos using Gigwalk for panoramic 3-D photosynthesis of businesses and restaurants in Bing Map. mCrowd developed by University of Massachusetts Amherst supports a feature of

¹ <http://www.innocentive.com/>

² <https://www.mturk.com/mturk/welcome>

³ <http://crowd.cs.umass.edu/>

⁴ <http://www.fieldagent.net/>

⁵ <http://gigwalk.com/>

interfacing with existing labor marketplaces such as Amazon's M-Turk and ChaCha in which location agnostic tasks can be effectively handled.

Given that location-based ubiquitous crowdsourcing is still in its early stage, understanding the user experiences is very important. A recent study on the user experiences reports the following observations (Alt, et al., 2010). People prefer address-based task selection (or map-based visualization). Among all the tasks, photo taking tasks were most popular mainly due to lower cognitive overhead. People are not willing to make significant efforts on performing tasks as most completed tasks are close to their usual places (e.g., home and work). When searching for tasks, people wish to view the tasks close to their current location. Tasks are usually done after work or during breaks, and the response time varies widely (several minutes to one day or longer). Voluntary tasks (with no monetary rewards) have much lower chances of completion rate as opposed to tasks with monetary rewards.

5.2 Human Intelligence and Social Networking

We review ubiquitous knowledge services that exploit crowdsourced sensing and collaboration over social networking tools such as Twitter and Foursquare. Several application services have proposed by the researchers at University of Buffalo, e.g., weather information collection and noise mapping (Dermirbas, et al., 2010) and location-based query resolution (Fatih, Yilmaz, and Dermirbas, 2011).

Given that the current mobile platforms lack an infrastructure that assists mobile users to perform collaboration and coordination ubiquitously and permits searching/aggregating the data published by mobile users, Dermirbas, et al. (2010) proposed a crowdsourcing architecture based on Twitter, an open social networking system that allows users to share short messages (140 character limit). The key components include Sensweet and Askweet. Sensweet is a smartphone application that uses the smartphone's ability to work in the background for sensing ambient environments without distracting a mobile user. Sensweet also defines a markup language called TweetML to standardize the sensor data format in Twitter. Askweet is a server program that monitors a Twitter account (dedicated for a specific service) for questions, processes the questions, and aggregates the replies to answer the questions. It first attempts to answer the queries using the data available on Twitter (including the data published by Sensweets). If that fails, Askweet finds experts on Twitter to which the queries are forwarded. Dermirbas, et al. (2010) demonstrated two application services: crowdsourced weather radar and noise mapping. In the crowdsourced weather radar service, random Twitter users are probed with a question about the current weather; in the noise mapping service, a Sensweets mobile application measures the noise level using the microphone and reports the results via Twitter. In their experiment of collecting responses of NYC weather, they found that weather probing takes reasonable delay (e.g., more than 50% of answers have less than 30 minutes), and Twitter users are more responsive during afternoon and evening.

This work was extended to location-based queries by incorporating the location based social networking services, e.g., Foursquare (Fatih, Yilmaz, and Dermirbas, 2011). Search queries typically fall into two categories: factual, e.g., "hotels in Miami" and non-factual, e.g., "cheap, good hotel in Miami." In the case of non-factual queries, it is well known that traditional search engines perform poorly. Noting that significant fraction of location-based queries is non-factual, Fatih, Yilmaz, and Dermirbas extended the aforementioned system by considering the location information of users (from location-based social networking services such as Foursquare) when selecting people to ask questions. Their experimental results showed that the user participation significantly improved when location information was used; e.g., the median response delay was 13 minutes.

Instead of asking Twitter users to perform human computation tasks, Sakaki, Okazaki, and Matsuo (2010) showed that Twitter messages posted by users can be automatically analyzed to detect natural events such as earthquakes and typhoons. They proposed a machine learning based approach using support vector machines for event detection. For earthquake detection, features of interest include statistics (e.g., tweet message length), key words (e.g., shaking, earthquake), and word context (e.g., location, time). Like real sensors, semantic analysis may result in errors. For accurate detection, time-series data are further analyzed to build spatial/temporal models; e.g., the frequency of earthquake related messages exponentially decreases. In the experiments, they analyzed total 49,314 tweets over the one year period from August 2009 to September 2009 (Sakaki, Okazaki, and Matsuo, 2010). When compared the results with Japan Meteorological Agency (JMA) data, they were able to detect 96% of earthquakes that were stronger than scale 3 or more during the period.

6 Building Ubiquitous Knowledge Service Platforms

Accessing remote sensor networks (e.g., sensor data retrieval) is one of the critical requirements of ubiquitous knowledge services. If there is only a single sensor network, we can assign a single node as a sink (or a base station or gateway) for data collection. When there are multiple wireless sensor networks that are geographically distributed, diverse system architectures for data collection are feasible. A simple approach would be using a centralized server over the Internet. An alternative is to maintain a hierarchical structure; e.g., local servers maintain all the data generated in the local area, and a central server keeps the index for global data retrieval, or to use peer-to-peer storage; e.g., distributed hash tables maintain all the sensor data.

So far various Internet-based systems were proposed for these purposes; e.g., a simple two-tier structure (local vs. backend servers) such as ArchRock (Woo 2007), SensorBase (Reddy, et al., 2007), and SensorMap (Nath, Liu, and Zhao, 2007), a semi-hierarchical structure such as IrisNet (Gibbons, et al., 2003), and Global Sensor Networks (GSN) (Aberer, Hauswirth and Salehi, 2006), and a fully distributed peer-to-peer structure such as GeoServ (Ahn, Lee, Moon, 2011). In ArchRock and SensorBase, sensor data from a sensor network is aggregated at the local gateway and is published to the front-end server through which users can share the data. In SensorBase, back-end servers (called re-publishers) further process sensor data to enable sensor data searching. SensorMap is a web portal service that provides mechanisms to archive and index data, process queries, and aggregate and present results on geocentric Web interfaces such as Microsoft Virtual Earth. In IrisNet, each organization maintains database servers for its own sensors, and a global naming service is provided for information access; a similar approach is used in GSN to allow users to query local and remote sensor data sources. GeoServ, a fully-distributed peer-to-peer sensor networking platform based on a distributed hash table, provides a location-aware sensor data retrieval scheme for geographic range queries, and a location-aware publish-subscribe scheme for multicast routing to a group of subscribed users (Ahn, Lee, Moon, 2011).

Along with these research prototypes, there have been continued efforts on standardization of technologies such as Semantic Web Enablement (SWE), Semantic Sensor Web (SSW), machine-to-machine (M2M) communications, and EPCglobal Network. Also, several related research projects were recently conducted in Europe such as BRIDGE (BRIDGE, 2009) and SENSEI (SENSEI, 2010). These standards and research systems feature various functions ranging from simple sensor data collection to inter-networking across multiple domains to semantic sensor data analysis. In this section, we review these activities and provide a detailed comparison of key features. Further, we present our research proposal called the knowledge communications service (KCS).

6.1 M2M

In the past, the communication system mostly considered the scenarios with human interactions, namely human-to-human (H2H) such as voice communications and human-to-machine (H2M) such as remote service access including web browsing. Recently, the industry has been also looking at alternative communication scenario where human interactions are absent, which is called machine-to-machine (M2M) communication or machine type communication (MTC). A service optimized for M2M differs from a service optimized for H2H communications in that it has different market scenarios with a potentially very large number of low-cost sensor nodes with little data traffic per node (3GPP, 2010a) (3GPP, 2010b). Service areas of M2M include security, tracking/tracing, payment, health, remote maintenance control, metering, and consumer devices. For example, in the smart grid area, autonomous and smart monitoring with M2M communications enables utility providers to remotely sense and control their utility equipment, thus obviating the need of laborious manual intervention. Many analysts expect that the M2M market will increase drastically in the near future.

Many groups have actively involved in M2M standardization, e.g., IEEE 802.16p (WiMAX), IEEE802.11 (Wi-Fi), 802.15.4 (ZigBee), GSMA, WiMax Forum, WFA, OMA, TTA, CCSA NITS, etc. Here, we overview the proposals of leading standardization groups, namely 3GPP and ETSI. 3GPP has considered M2M as one of the most promising applications to support in the LTE-Advanced standards (3GPP, 2010b). For successful adoption, 3GPP proposed to improve their cellular network systems as follows (3GPP, 2011): (1) architectural enhancements to support a large number of machine type communications (MTC) devices in the networks; (2) architectural enhancements to fulfill MTC service requirements; and (3) support combinations of architectural enhancements for MTC. While 3GPP focused on communications networks, ETSI aims at standardizing more generic service architecture and

requirements of M2M (ETSI, 2010)[4]. It claimed the following clauses: (1) general requirements describing communications features necessary for the correct establishment of M2M communications; (2) management related to malfunction detection, configuration, and accounting; (3) functional requirements for M2M services such as data collection/reporting and remote control operations; (4) security for M2M device such as authentication, data integrity, and privacy; and (5) naming, numbering, and addressing intended to M2M devices. ETSI also provided the overall end-to-end M2M functional architecture, including the identification of the service entities, the related interfaces, and the relations with the underlying existing networks (ETSI, 2011).

The following are the core system components of ETSI proposals (see Figure 8): (1) M2M devices and local area networks, (2) M2M gateway, (3) M2M network core, and (4) M2M applications. M2M applications interact with the service capabilities in the M2M network core to acquire the sensor data of interest. Actual sensor data will be collected via the M2M area network through M2M gateways (or sink nodes). Any local service capabilities in the M2M gateways are registered at the network core for centralized access and control of service capabilities in the M2M network.

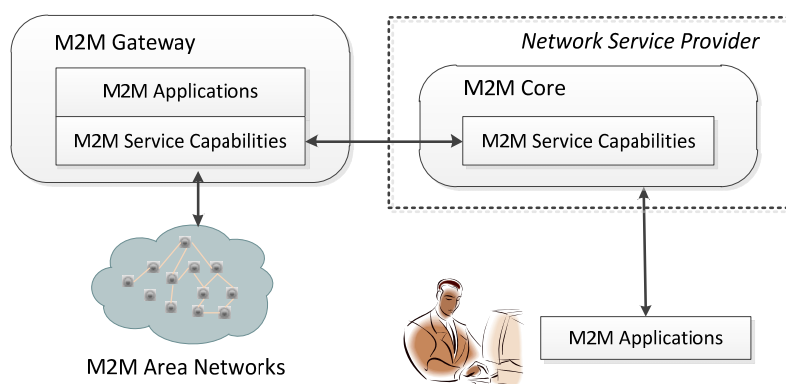


Figure 8 M2M architecture overview

6.2 Sensor Web Enablement (SWE) and Semantic Sensor Web (SSW)

Sensor Web Enablement (SWE) is an initiative of the Open Geospatial Consortium (OGC) to add real-time sensing to the Internet and the Web, by building an open standard framework for exploiting Web-connected sensors and sensor systems, e.g., flood gauges, Webcams, etc. (Botts, Percivall, Reed, and Davidson, 2008). SWE has been focused on developing Web standards (based on XML) to enable (1) discovery, (2) access, and (3) tasking, and (4) alert notification. SWE's standards can be classified into two categories: (1) encoding standards: how to describe sensors and sensor data (i.e., SensorML and O&M), and (2) Web services standards: how to provide sensor Web services (i.e., SOS, SPS, and SAS). Sensor Model Language (SensorML) describes the standard models and XML schema for describing the detailed information about sensors and their processes (e.g., functional models and process chains), including the information needed for sensor discovery. Observations and Measurement Schema (O&M) defines the standard models and XML schema for encoding sensor data (e.g., units, features). Sensor Observations Service (SOS) provides a standard web service interface for querying and retrieving sensor data and sensor system information. Sensor Planning Service (SPS) is a standard web service interface for configuring the sensors (and testing sensing feasibility), and Sensor Alert Service (SAS) is a standard web service interface for publishing and subscribing to alerts from sensors. As shown in Figure 9, for a given set of sensors the service catalogs (i.e., the registries of the sensor system) are built based on SensorML and O&M, and the resulting catalogs are used in SPS and SOS (e.g., discovering services, sensors, providers, and sensor data). Once the sensing system has initialized, a user can make a collection request to SPS or directly to the sensors (via SOS).

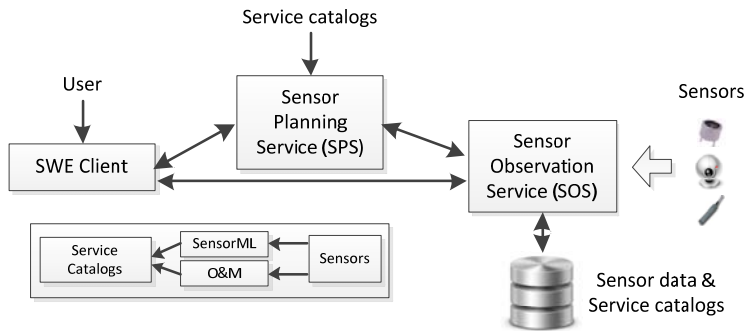


Figure 9 SWE overview

The Semantic Sensor Web (SSW) is a framework for providing “enhanced meaning to sensor data” to improve situation awareness by augmenting standard sensor languages of the SWE with semantic annotations through which applications can understand and reason over sensor data consistently, coherently, and accurately (Sheth, Henson, and Sahoo, 2008). SSW is based on the semantic Web, an evolving extension of the Web where semantics (meaning) of information is captured using ontologies, enabling machines to interpret and relate data. As shown in Figure 10, the key components of the semantic Web include the Resource Description Framework (RDF) data representation model, and the ontology representation languages, namely RDF schema, and Web Ontology Language (OWL). For semantic annotation, SSW uses spatial, temporal, and thematic ontologies along with domain-specific ontologies for representing sensors (e.g., biomedical ontology). Here, annotation spans from raw sensor data to extracted features and detected entities from sensor data. By using Semantic Web tools, semantic reasoning and computation over semantically annotated data can be performed, including semantic enabled search, integration, answering complex queries, pattern finding, discovery, visualization, etc. For instance, if a group of sensors provides information about temperature and precipitation, possible road condition can be specified using simple rules (say, an icy road if temperature is below 0 degree Celsius and it is raining).

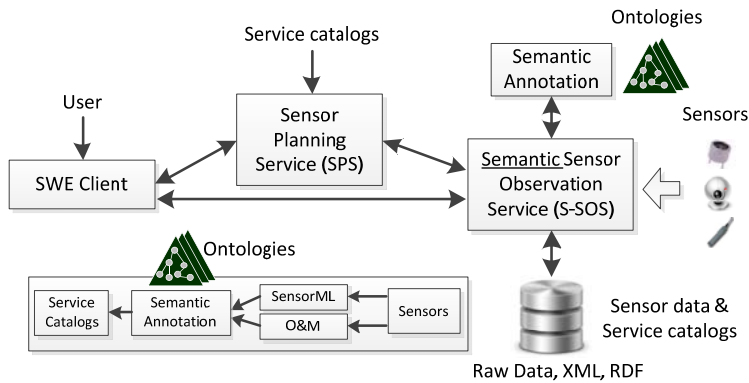


Figure 10 SSW overview

6.3 Internet of Things

The phrase Internet of Things (IoT) was coined by the Auto-ID Center with the vision of building a global network called the EPC network that enables tracking items (e.g., using RFID tags and readers) and sharing information over the Internet. Due to recent advances of RFIDs, sensor networks, near field communications (NFC), and smart devices, the Internet of Things has been widely accepted as an enabler of ubiquitous sensor networking over the Internet; i.e., remotely monitoring the physical world using sensors (or associated services) and acting upon changes in the real world (Santucci, 2010). In fact, such real-time sensing and actuation call for open, scalable, secure infrastructure standards. Some of the recent efforts include Auto-ID’s EPC Network (and EPC sensor networks), OCG’s Sensor Web Enablement (SWE), ISO/IEC standards, and IEEE 1451 standards (Song, and Lee, 2008).

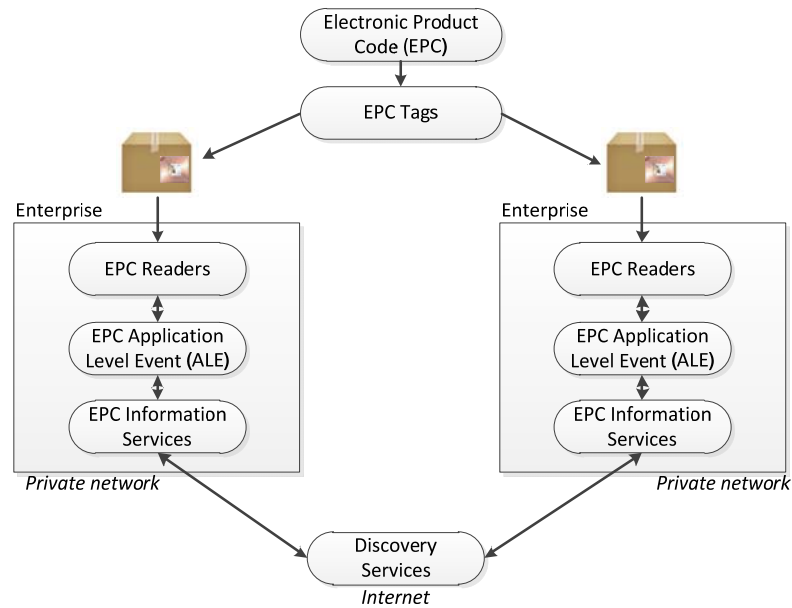


Figure 11 EPC Network overview

The EPC network basically considers small, inexpensive, high performance RFID tags and readers. Its goal is to provide the ability to provide item tracking and information sharing to the current supply chain architecture. In the EPC network, all items are uniquely identified using Electronic Product Code (EPC). Information related to each item is stored within a network of databases with each touch point within the supply chain collating information specific to that product. The EPC network also defines a common communication language and standard interfaces, thus guaranteeing interoperability between vendors. As shown in Figure 11, the EPC network is composed of the following components: Electronic Product Code (EPC), ID System (RFID tags and readers), EPC Application Level Event (ALE), EPC Information Services, and Discovery Services. EPC is an identification scheme for uniquely identifying physical objects. The current coding schemes are based on the GS1 family of codes, but it is possible to use any existing coding schemes. An RFID tag with the EPC code will be attached to an object, and RFID readers can read out information. EPC Application Level Event (ALE) specifies how the captured RFID data are filtered and grouped to some events. After passing through the EPCALE, the captured RFID data will be stored into the EPCIS's repository. The querying interface of EPCIS allows external applications to access the RFID data. Discovery services enable global access to data associated with a specific EPC (e.g., between different enterprises).

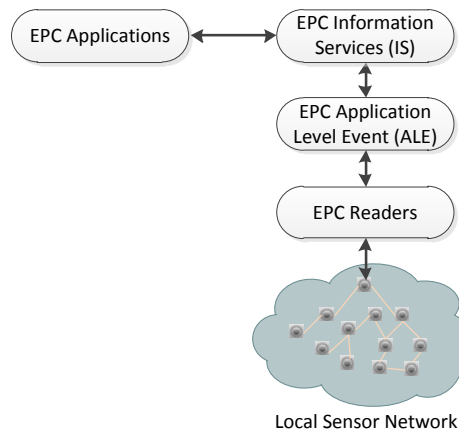


Figure 12 EPC Sensor Network

While the EPC network considers passive RFID tags, it can be also extended to include smart tags with communication and processing capabilities (forming an EPC sensor network) (Sung, Sánchez, and Kim, 2007). As shown in Figure 12, a group of smart tags (or a local sensor network) is connected to the backend system via a reader (called a base station). It uses IEEE 1451 TEDS (Transducer Electronic Data Sheet) standards for sensor and sensor data description. Various air interfaces are used for communications among smart tags (e.g., Bluetooth, ZigBee, and NFC), but it does not specify sensor networking protocols in a local sensor network. Unlike the original EPC network, the functions of EPCALE are extended to support more complex operations on sensor data (e.g., data processing such as averaging).

Like the EPC Sensor Network proposal, the BRIDGE (Building Radio Frequency IDentification for the Global Environment) project aimed at supporting ambient sensors and sensor-enabled active RFID tags in the EPCglobal network for supply chain monitoring (BRIDGE, 2009). Given that existing RFID related standards and proposals cannot efficiently support sensor data, BRIDGE incorporates OGC's Sensor Web Enablement (SWE) into the EPC Network. These standards share the same functionalities; e.g., EPC Network's EPCIS/ALE corresponds to SWE's SOS and Discovery Service (ONS) to Catalog. Yet SWE lacks the concept of base station (BS), reader management, and globally unique address, while EPC Network does not provide sensing task planning. In BRIDGE, SWE is extended to include the base station and to inter-work with the EPC Network, which causes minimal modification to the EPC Network standards.

The SENSEI project models wireless sensor and actuator networks into resources and proposes to network "resources" over the Internet (SENSEI, 2010). For instance, a sensor node in the vehicle can provide one or more resources (e.g., engine and speed sensors). SENSEI offers a rendezvous service to allow locating resources of interest (called a resource directory). Moreover, it supports components that enable contextualized information retrieval and interaction, which requires an entity directory. Sensor data are represented based on the O&M standards and can be processed to derive entity-based contextual information. SENSEI is composed of the following components: Resource Directory, Entity Directory, Execution Manager, and Semantic Query Resolver. A user's query is routed to the semantic query resolver, which then access the entity/resource directories to make execution plans. The resulting plans are then submitted to the execution manager, and the final results will be delivered to the user.

6.4 Feature Comparison

The key features of knowledge integration and retrieval systems can be summarized as follows: sensor model and encoding, local area sensor networking, application service support, dynamic multi-domain service networking, semantic query processing, service-device/network interaction, human computation/sensing, social networking, and agent support. The feature comparison results can be found in Table 1. In SWE, sensor model and encoding is supported via O&M and SensorML, and application services are supported via SOS and SPS. The EPC Network was designed to support RFID tags and related services, including multi-domain service networking (e.g., different enterprises). The EPC Sensor Network and BRIDGE extend the EPC Network to support active sensors. SENSEI provides "resource" (or service) based networking and semantic query processing. ESTI's M2M standards basically provide similar features as the EPC Sensor Network and BRIDGE, but the current M2M standards focused on single domain scenarios. .

From our review on smart devices and related mobile/people sensing applications, including ubiquitous crowdsourcing, we find the following observations: (1) humans are part of the systems (human computation), (2) existing mobile sensing services are usually optimized due to resource constraints (requiring service-device/network interactions), and (3) when human resources are utilized for knowledge services, exploiting social networks is crucial. Another relevant technique is agent technologies that can automatically collect user information and infer a user's wants and needs to provide personalized services. In the following, we briefly introduce our proposal called the Knowledge Communications Service (KCS) that provides these features by interworking with sensors, smart devices, and people.

	SWE	SSW	EPC-NET	EPC-SN	BRIDGE	SENSEI	M2M ESTI	KCS
RFID support			○	○	○	○	○	○
Sensor model and encoding	○	○		○	○	○	○	○
Local area sensor networks				○	○	○	○	○
Service support	○	○	○	○	○	○	○	○
Dynamic multi-domain service networking			○	○	○	○		○
Semantic query processing		○				○		○
Service-device/network interaction								○
Human computation (and sensing)								○
Social networking								○
Agent support								○

Table 1 Feature comparison

6.5 Towards Knowledge Communications Services

The goal of knowledge communications service (KCS) is to perceive the intention of a user and provide the customized knowledge services to the user at any place and any time, with any device even with physical restrictions. To attain high quality user experiences, exploiting device/network, application, and human intelligence is necessary. We envision new system architecture for such ubiquitous knowledge services with the following components (see Figure 13): Knowledge Server (KS), Knowledge Gateway (KG), and Smart Device (SD). The desired features of the KS include knowledge brokering, virtual user knowledge agents, semantic query processing, and knowledge repository. Knowledge brokering enables content exchanges (e.g., data, information, and knowledge) across KSs in different network/domains (e.g., Verizon, AT&T, and KT) and across various content providers (e.g., Google, Yahoo!, Naver). The collected data will be stored in a KS's knowledge repository for later access. Virtual user knowledge agents and semantic query processors will permit personalized knowledge services; i.e., the user agent will proactively search for content with help of the semantic query processors that gather and analyze all the available data distributed over the network. A KS interacts with knowledge gateways (KGs) in the same domain. A KG has a light-weight KS (e.g., with knowledge brokering, agents, and caching) and supports inter-networking with legacy wireless sensor networks including M2M and RFID tags. Smart devices (SDs) are equipped with a user knowledge agent, data/information gathering units, and adaptive/flexible user interfaces for efficient human-computer interactions. User knowledge agents basically use the services in a local KG and central KSs for knowledge acquisition and processing. Further, they can use direct device-to-device communications among SDs (including with KGs) to acquire resources from nearby nodes (e.g., data, information, knowledge, and computing power). In this way, the agent can collect and process a user's context information to accurately predict the intension and to provide personalized knowledge services.

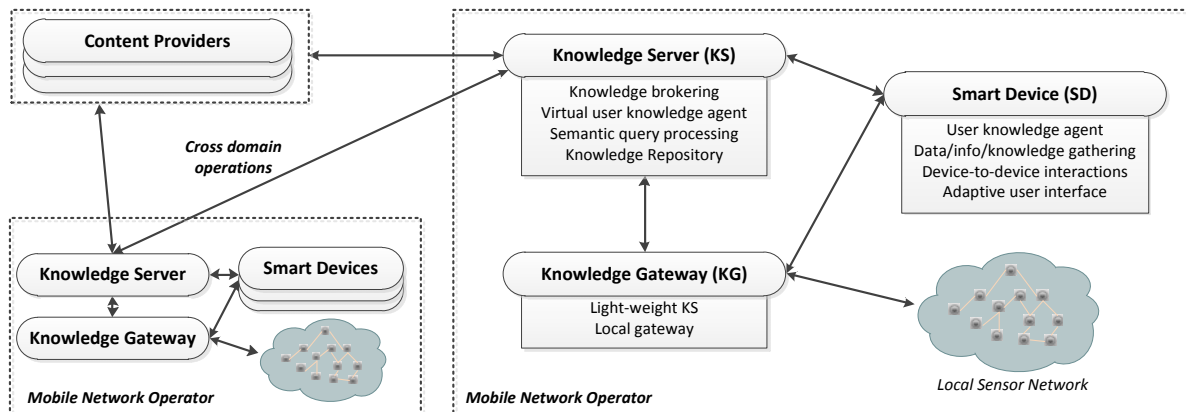


Figure 13 Knowledge communications service overview

7 Conclusion

We introduced the concept of ubiquitous knowledge services that seamlessly integrate content (e.g., data, information, knowledge) from various sources (e.g., content providers, sensors, smart devices, humans) at large scales and derive new values for end users. In this chapter, we began with reviewing the key enabling technologies of ubiquitous knowledge services in the areas of device, network, application, human intelligences. We then reviewed and compared the existing platforms for sensor data collection, retrieval, and application services. Finally, we proposed a set of desirable features of the ubiquitous knowledge service platforms based on our vision of the knowledge communications service (KCS).

8 References

- 3GPP. "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN)," TS 36.300 V10.0.0, 2010.
- 3GPP. "Service requirements for machine-type communications; Stage 1," TS22.368 v.10.1.0, 2010.
- 3GPP. "System improvements for machine-type-communications," TS23.888 v.1.4.0, 2011.
- Aberer, K., M. Hauswirth, and A. Salehi. "A Middleware for Fast and Flexible Sensor Network Deployment," In *VLDB*, 2006.
- Ahn, J. H., U. Lee, H. J. Moon. "GeoServ: A Distributed Urban Sensing Platform," In *IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid'11)*, 2011.
- Alt, F., A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz. "Location-based Crowdsourcing: Extending Crowdsourcing to the Real World," In *NordiCHI*, 2010.
- Botts, M., G. Percivall, C. Reed, and J. Davidson. "OGC® Sensor Web Enablement: Overview and High Level Architecture," In *ISCRAM*, 2008.
- BRIDGE. "Sensor Based Condition Monitoring," BRIDGE Consortium, 2009.
- Bulut, M. F., Y. S. Yilmaz, and M. Demirbas. "Crowdsourcing Location-based Queries," In *Percom Workshops*, 2011.
- Burkhard, R. "Learning from Architects: The Difference between Knowledge Visualization and Information Visualization," In *International Conference on Information Visualization*, 2004.

- Burkhard, R., and M. Meier. "Tube Map: Evaluation of a Visual Metaphor for Interfunctional Communication of Complex Projects," In *I-KNOW*, 2004.
- Demirbas, M., M. A. Bayir, C. G. Akcora, and Y. S. Yilmaz. "Crowd-Sourced Sensing and Collaboration Using Twitter," In *WOWMOM*, 2010.
- Dikaiakos, M. D., S. Iqbal, T. Nadeem, and L. Iftode. "VITP: an Information Transfer Protocol for Vehicular Computing," In *VANET*, 2005.
- Douglis, F., R. Caceres, B. Marsh, F. Kaashoek, K. Li, and J. Tauber. "Storage Alternatives for Mobile Computers," In *USENIX*, 1994.
- EPCglobal Inc. "The EPCglobal Architecture Framework," 2009
- Eppler, M., and R. Burkhard. *Knowledge Visualization*. Encyclopedia of Knowledge Management, Idea Group, 2005.
- Eriksson, J., L. Girod, B. Hull, R. Newton, H. Balakrishnan, and S. Madden. "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," In *MobiSys*, 2008.
- ETSI TS 102 689 V1.1.1. "Machine-to-machine Communications (M2M); M2M service requirements, 2010.
- ETSI TS 102 690 V0.13.3. "Machine-to-machine Communications (M2M); Functional architecture, 2011.
- Frenkiel, R., B. Badrinath, J. Borras, and R.D. Yates. The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data," *IEEE Personal Communications* 7-2 (2002): 66-71.
- Ganti, R. K., N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. "GreenGPS: A Participatory Sensing Fuel-efficient Maps Application," In *MobiSys*, 2010.
- Gibbons, P. B, B. Karp, Y. Ke, S. Nath, and S. Seshan. "IrisNet: An Architecture for a Worldwide Sensor Web," *IEEE Pervasive Computing*, 2 (2003): 22-33.
- Hull, B., V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden. "CarTel: a Distributed Mobile Sensor Computing System," In *SenSys*, 2006.
- Jiang, D., V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, Design of 5.9 GHz DSRC-based Vehicular Safety Communication," *IEEE Wireless Communications* 13 (2006): 36-43.
- Khatib, M. G., B.-J. van der Zwaag, P. H. Hartel, and G. J. M. Smit. "Interposing Flash between Disk and DRAM to Save Energy for Streaming Workloads," In *ESTIMedia*, 2007.
- Koukoumidis, E., L.-S. Peh, and M. Martonosi. "SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory," In *MobiSys*, 2011.
- Lee, U., E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi. MobEyes: smart mobs for urban monitoring with vehicular sensor networks, *IEEE Wireless Communications* 13 (2006): 51-57.
- Lee, U., S. Jung, A. Chang, D.-K. Cho, and M. Gerla. "P2P Content Distribution to Mobile Bluetooth Users," *IEEE Transaction on Vehicular Technology*, 59 (2010): 356-367.
- Lu, H., W. Pan, N. D. Lane, T. Choudhury, A. T. Campbell. "SoundSense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones," In *MobiSys*, 2009.
- Mell, P., and T. Grance. "The NIST Definition of Cloud Computing," National Institute of Science and Technology, 2011.

- Miluzzo, M., N. D. Lane, K. Fodor, R. A. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. "Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application," In *SenSys*, 2008.
- Mohan, P., V. Padmanabhan, and R. Ramjee. "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," In *SenSys*, 2008.
- Mun, M., S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M.H. Hansen, E. Howard, R. West, and P. Boda. "PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research," In *MobiSys*, 2009.
- Nath, S., J. Liu, and F. Zhao. "SensorMap for Wide-Area Sensor Webs," *IEEE Computer Magazine*, 40 (2007).
- Oviatt, S. L. "Advances in Robust Multimodal Interface Design." *IEEE Computer Graphics and Applications*, 23 (2003): 52-68.
- Oviatt, S. L. "Multimodal Interactive Maps: Designing for Human Performance," *Human-Computer Interaction*, 12 (1997): 93-129.
- Quinn, A. J., and B. B. Bederson. "Human Computation: A Survey and Taxonomy of a Growing Field," In *CHI*, 2011.
- Reddy, S., G. Chen, B. Fulkerson, S. J. Kim, U. Park, N. Yau, J. Cho, M. Hansen, and J. Heidemann. "Sensor-Internet Share and Search—Enabling Collaboration of Citizen Scientists," In *DSI*, 2007.
- Sakaki, T., M. Okazaki, and Y. Matsuo. "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors," In *WWW*, 2010.
- Santucci, G. "The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects," 2010.
- SENSEI. "The SENSEI Real World Internet Architecture," Sensei Consortium, 2010.
- Sheth, A., C. Henson, and S. S. Sahoo. "Semantic Sensor Web," *IEEE Internet Computing*, 8 (2008): 78-83
- Song, E. K., and K. Lee. "Understanding IEEE 1451 Networked Smart Transducer Interface Standard," *IEEE Instrumentation & Measurement Magazine*, (2008).
- Sung, J., L. T. Sanchez, and D. Kim. "The EPC Sensor Network for RFID and WSN Integration Infrastructure," *PerCom Workshops*, 2007.
- Thiagarajan, A., J. Biagioni, T. Gerlich, and J. Eriksson. "Cooperative Transit Tracking using Smart-phones," In *SenSys*, 2010.
- Thiagarajan, A., L. Ravindranath, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, H. Balakrishnan. "VTrack: Accurate, Energy-aware Traffic Delay Estimation Using Mobile Phones," In *SenSys*, 2009.
- Weiser, M. "The computer for the 21st century." *ACM SIGMOBILE Mobile Comput Commun Rev* 3 (1999): 3-11
- Woo, A. "A New Embedded Web Services Approach to Wireless Sensor Networks," In *SenSys*, 2007.
- Yoon J., B. Noble, and M. Liu. "Surface Street Traffic Estimation," In *MobiSys*, 2007.