# Wireless Router as a Physical Access Control System (WRPACS)

Dragos George Comaneci, Silvia Cristina Stegaru, Ciprian Dobre
Automatic Control and Computers Faculty
Politehnica University of Bucharest
Emails: {dragos.comaneci, silvia.stegaru}@cti.pub.ro, ciprian.dobre@cs.pub.ro

***Abstract*—Smartphones today integrate many sensors and provide large computing capacities. They enable the shift towards massive quantities of real-time information becoming access push rather than demand pull on a global case. CAPIM, a platform to support such a paradigm, integrates services to monitor and a context for adapting with the user's context using the sensors and capabilities of smartphones, together with online social data. It integrates context-aware services that are dynamically configurable and use the user's location, identity, preferences, profile, and relations with individuals, as well as capabilities of the mobile devices to manifest themselves in many different ways and re-invent themselves over and over again. In this paper we present the design and development details of the security and user identification components to support these services. We propose a secure platform for user authentication and session management, based on public key infrastructure (PKI) services. We analyze its strengths and weaknesses, and present as a case study the particular extension of the platform to support secure user access to restricted areas of a building. We also discuss an analysis of the implementation, cost assessments and problems that might arise, as a methodology to support the construction of such mobile and context-oriented applications.**

***Index Terms*—wireless router, access control, microcontroller, magnetic lock, smartphone**

## I. Introduction

As people realize that having more sensing and computing capabilities in every-day situations is attractive for many reasons, smartphones become commodity hardware. Their success is the basis for a shift towards developing mobile applications that are capable to recognize and pro-actively react to user's own environment. Such context-aware mobile applications can help people better interact between themselves and with their surrounding environments. This is the basis for a paradigm where the context is actively used by applications designed to take smarter and automated decisions: mute the phone when user is in meeting, show relevant information for the user's current location, etc. CAPIM (Context-Aware Platform using Integrated Mobile services) [4] is a solution designed to support the construction of context-aware applications. It integrates services designed to collect context data (location, user's interests and characteristics, as well as the environmental data) and use it to provide a richer and simpler experience for the end user.

In the present work we provide an implementation of a fundamental part of CAPIM's design considerations for the management of user identity in context-aware applications. The user's identity is required by many context models. It can be used to infer preferences that are actively used in favor of the user, or it is used to provide personalized sets of services. Today Public Key Infrastructures (PKI) solutions are generally accepted to support the management of identity. PKI provides a standardized and legally recognized service support [2]. Therefore it makes sense to use such services in providing electronic identity in context-aware integrated mobile services. PKI provides services such as confidentiality, integrity, authentication and non-repudiation [10]. By using the standards defined by PKI we can develop an approach to support the construction of rich context-aware applications that use the identity of the user as active context information. In particular, the security layer is used from the construction of social-aware mobile applications to intelligent housing, capable of actively recognizing the user entering the room for example. As such, in our current paper we shall focus on a mechanism for secure access to a physical area of a building built on top of CAPIM that leverages off the shelf hardware components (such as a wireless router) in order to provide the required services.

The rest of the paper is structured as follows. Section 2 presents similar projects already implemented to secure user access using a mobile handset. Section 3 shows an overview of the architecture of the system with a short description of all the main components and their role. Section 4 presents some implementation details of the core services present on the wireless router. Section 5 deals with test scenarios, possible system vulnerabilities and results as well as system deployment issues and costs. In Section 6 we conclude our discussion and present future work.

## II. Related Work

A similar notable project in the area of secure user access to restricted areas of a building with the use of a mobile handset has been developed at the Disco Lab at Rutgers University [10], [8]. Although the approach is different from our own (both in technology and system design), the goal is similar: to allow the use of a mobile handset as an electronic key, or, more generally, and electronic ID in order to access distributed services. However, we provide a more generic platform that actually include context as part of the entire process. Our proposed platform provides security guarantees as to who is accessing the contextual services, where people are. It instruments using context-oriented policies the interactions between peoples and services. In particular, we present a case study for the use of the platform as a tool to create a simple instrument to mediate the access for the user inside an intelligent building, capable of recognizing the user.

## III. Architecture

The proposed system requires the interaction of many different components. These components can be summarized into two main parts : the Secure Service Communication Platform (SSCP) and the Secure Area Access Service (SAAS) that is built on top of SSCP.

### *A. The Secure Service Communication Platform (SSCP)*

The main components of SSCP are presented in Figure 1. On the mobile side two services are executed: the CAPIM Secure Communication Service, and the User Security Service. The CAPIM Secure Communication Service is responsible for session establishment, as well as for communication with other services that require user identification. The User Security Service implements the PKI operations for loading the user certificate, establishing of an SSL context for example, etc.

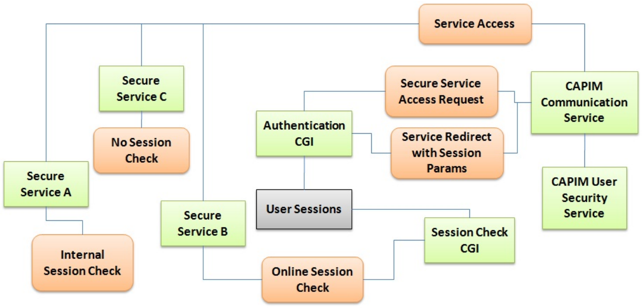

Figure 1. SSCP main components.

On the server side of the platform there are main two components responsible for authentication and authorization. The authentication CGI (Common Gateway Interface) component is responsible for verifying the credential of the user, for creating the appropriate session and registering it in a local database. The session check CGI component, as its name suggests, is used by the other services as an interface to check validity and retrieve information related to the user's context (such as the user's rights as defined by the security policy).

For more fine-grained audit requirements, the sessions (verifications and policy rights) can be established per service. In this case the communication service on client-side, upon authentication, also submits the name of the service that it requires access to. The authentication CGI verifies the name of the service in the local database, retrieves a certificate associated with that service, and uses the associated public key to encrypt a hash of the session key.

As illustrated in Figure 1, the services have three options for checking the user session, depending on the service requirements. The first option, used in the figure by service A, is an internal check of the session key. The session key is signed by the authentication service with its private key, so other services that have the certificate of the authentication service can use the public key stored within it to check the signature. This approach for verification is fast, but it involves processing on the service receiving the session key. Also, not much information about the user can be stored within the session key (which should be small, because it is transmitted with each request the user makes for a service). Therefore, this approach is suitable for services that only require a valid user and no other information (for example security rights).

The second option of verification, used by service B in Figure 1, is online session check. In this case the service receiving the session key from the user establishes a connection to the session check CGI. The session check CGI looks up the session in its' local database, retrieves the information associated with it (such as user details, user rights, etc.) and sends this information back to the requesting service. This approach involves minimal work on the service side but is also much slower since it requires communicating with the session check CGI. Another advantage is that the service can retrieve bundles of information associated with the user.

The third option (provided for consistency) is for services that do not require user identification. In this case the service simply ignores the session key.

### *B. The Secure Area Access Service*

Access to a secure area of a building has always been a constant security problem and a lot of specialized solutions have been developed to facilitate this service [6], [1], [11]. Still, today they are either impractical for the user or lack the necessary security required for accessing sensitive areas. Also, the costs, both in equipment and training involved in implementing some of the solutions are prohibitive.

The approach proposed in CAPIM is feasible because many users today carry at least one smartphone. The idea is to have a key in the form of a digital certificate and associated private key stored on the mobile smartphone and, after authenticating through SSCP and getting a session key, use the obtained session key to send an access request for a certain area to the service.

In order for the solution to work, the mobile smartphone has to be connected to the local Wi-Fi network. This is required to access the authentication service. Still, because the average Wi-Fi communication range is tens of meters, a more location sensitive solution is needed to determine that the user is in the presence of a door that protects access to a secure area. Hence, we also need to use Bluetooth for our propose access service.

The proposed solution works as follow. In the beginning the users' mobile smartphone is connected to the local Wi-Fi network and authenticated through SSCP, thus having a valid session key. Using the location service the mobile handset determines that it is in the proximity of a door that leads to a restricted area, and automatically turns on the Bluetooth receiver on the phone and scans the area for devices. The mobile handset finds the device corresponding to the Bluetooth dongle of the door and proceeds to generating a random shared-key that will be used for the association between the two Bluetooth devices. The random shared-key is posted along with the MAC address of the mobile handset via Wi-Fi to the Secure Area Access Service (SAAS). The mobile handset then begins the Bluetooth association procedure with the dongle of the door. The device controlling the Bluetooth dongle of the door detects the MAC of the device trying to associate and queries the SAAS for the random shared-key generated by the mobile handset, retrieves it and uses it to carry out Bluetooth association. After the association is complete the mobile handset sends a hello message and the door is opened. The association between the two devices is kept for a limited amount of time (for example 1-2 hours) so further access through the door are simple. A visual representation of this process is presented in Figure 2.

To ensure a higher level of security for extra-sensitive areas, the SAAS may require the user to prove its identity. This is done using a biometric service based on face recognition [3], but this can easily be adapted to alternate biometric inputs, such as voice or fingerprint.

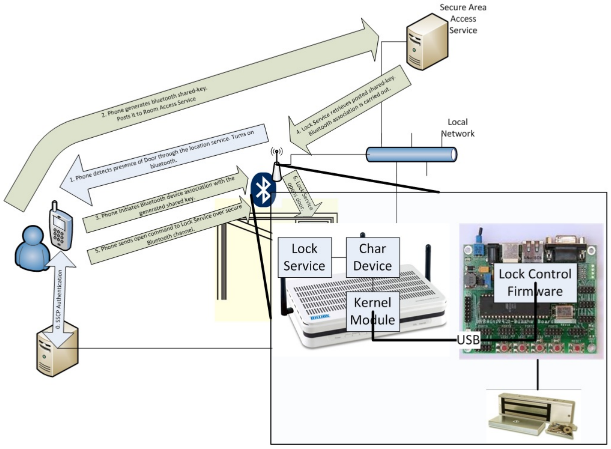
Secure Area
Access
Service
2. Phone generates bluetooth shared-key.
Posts it to Room Access Service
4. Lock Service retrieves posted shared-key.
Bluetooth association is carried out.
Local
Network
1. Phone detects presence of Door through the location service. Turns on bluetooth.
3. Phone initiates Bluetooth device association with the generated shared key.
5. Phone sends open command to Lock Service over secure Bluetooth channel.
6. Lock Service opens door.
0. SSCP Authentication
Lock
Service
Char
Device
Kernel
Module
Lock Control
Firmware
USB

Figure 2. Visual representation of the proposed solution.

Figure 2 also presents the components responsible for communication between the router and microcontroller board controlling the magnetic lock.

The main components of the Secure Area Access Context Service are presented in Figure 3. Emphasis is given to the location of each component in the system. The Lock Service daemon resides on the router controlling access to one or more restricted areas of a building. The daemon is responsible for monitoring Bluetooth connection requests from different Bluetooth dongle receivers connected via USB to router and also servicing door open requests for the doors it controls.

On the server side there is the Room Access Service, which is a Bluetooth shared-key repository where the mobile handset posts the randomly generated shared-key for Bluetooth association, along with the MAC address of the handset, the Lock Service present on the router following up and retrieving the posted key. Also on the server side the Location Service offers information regarding the location of access points to restricted area, and the Biometric Service which checks user biometrics and can be used in case of highly restricted areas.

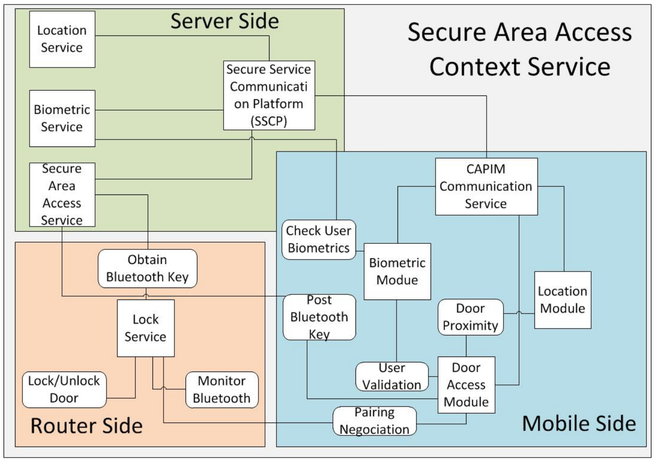

Figure 3. SAACS main components.

On the mobile side there is the Door Access Module, responsible for Bluetooth association, and the management of door access requests generated from door proximity alerts coming from the Location Module. Also, on the mobile side there is the Biometric Module, responsible for taking a photograph using the mobile handset integrated camera of the current user and sending it to the Biometric Service for verification.

## IV. Implementation

### A. Router side and server side

For the implementation and testing of the Lock Service an Asus 500gP V2 router with two USB ports was used. The firmware was replaced with the one provided by the open-source project DD-WRT [1] in order to have root access to the device and install the Bluetooth and door controller modules, along with the Lock Service. The Lock Service was developed in C++ and cross compiled for the MIPS32 platform to work with the processor present on the router. The Lock Service also uses the BlueZ Bluetooth library for accessing the dongle connected to the router and OpenSSL for accessing the Room Access Service.

A kernel module was also developed for communicating with the microcontroller board. It exposes a char device interface that can be used to both inspect the current status of the microcontroller and send open commands to the magnetic lock. The kernel module relies on the usbcore driver module and was compilled for the Linux 2.4 kernel because DD-WRT for Asus 500gP V2 only supports a 2.4.

An example of the hardware configuration is presented in Figure 4. The Room Access Service was developed, as in the case of SCCP, in C++ using FastCGI for communicating with the hosting web server. It stores the shared keys in a database and uses unixODBC for database access so that any flavor of database can be configured with it.

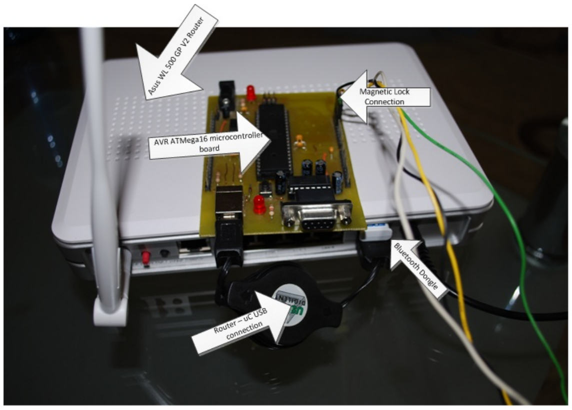

Figure 4. One possible hardware configuration.

The Biometric Service was developed in Java as a distributed service that supports multiple dispatchers and workers and uses the EigenFace image matching algorithm [7]. A more detailed description of the Biometric Service can be found in [3]. Details regarding the Location Service and Location Module, developed within the CAPIM project, are available in references [9], [5].

### B. Microcontroller board

In order to communicate with the magnetic lock the router was connected to a custom board through USB (as seen in Figure 4). The microcontroller used, Atmel ATmega16, was first flashed with AVRUSBBootloader (an USB bootloader for Atmel AVR controllers) to make it easier to program directly from the computer. The bootloader then loads our program at startup.

We have used the V-USB firmware for low-speed USB devices, modifying it to accept lock / unlock commands from the connected device, in our case the router. It then processes the command and acts accordingly, thus opening or closing the magnetic lock. First the V-USB microcontroller parameters were modified to work with our configuration. Secondly, we had to modify the driver signal interceptor so it would accept only the commands specified and ignore anything else received.

## V. Scenarios and Results

Being a complex system, several test scenarios have been developed in order to asses the overall security and possible attack vectors. This section presents different scenarios suitable for our system, as well as performance and reliability tests.

### A. Possible Security Attacks

Possible security breaches can stem from the technology in use as well as from the human element of the system, the mobile handset user. From a technology point of view, a possible attack can be expected in the form of an SSL man-in-the-middle attack. The man-in-the-middle attack (MITM), bucket-brigade attack, or sometimes Janus attack, is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, despite the fact that the entire conversation is controlled by the attacker. The attacker must be able to intercept all messages going between the two victims and inject new ones.

A man-in-the-middle attack can succeed only when the attacker can impersonate each endpoint to the satisfaction of the other, thus making it an attack on mutual authentication. In order to prevent this type of attack, SSL authenticates the server using a mutually trusted certification authority. As such, it is recommended that an internal private highly secured certification authority be used in order to generate the user and server certificates used within the system. An attacker, in order to apply the MITM attack would first need to obtain a certificate and associated private key that can be used as a server certificate from that specific certification authority. To prevent even the case in which an attacker manages to obtain such a certificate, we can construct a trusted certificate store containing all the certificates belonging to the services within the system and check this store during the SSL context establishment. Of course, in this case, some sort of update mechanism needs to exist in order to modify the certificate store in case of expiration or revocation. The certificate store update may be correlated with the CAPIM application updates.

The Bluetooth pairing mechanism can also constitute another attack vector but this implies that the attacker be able to listen in on the Bluetooth communication between the mobile handset and the locking mechanism. Also, since the pairing keys have a limited validity time, the attacker would need to intercept the pairing process once more in order to gain subsequent access to an area.

A denial of service attack (DOS) is also possible but can be easily prevented (at least on the server side components) through the use of the mod_evasive Apache DOS protection module.

Another possible attack would be for an attacker to be able to copy the user certificate and associated private key off the mobile handset and use as is. A completely secure solution to this problem can only exist if we can implement the equivalent of a PKI hardware token on the mobile handset. This can be done with the use of a specific component present on ARM processors dubbed TrustZone.

### B. Solution Cost Assesment

The project was designed to be as cheap to implement as possible so, for the server components, existing computing hardware may be used. The only costs incurred are for the routers, Bluetooth dongles, magnetic locks and magnetic lock controllers. A two USB port WiFi router can be found at a medium price of 60€. A Bluetooth dongle compatible with BlueZ incurs a cost of 20€. The magnetic lock controller can be built out of a microcontroller board with an USB port and two relays that control the locking mechanism, the total component costs for it being around 20€. The most expensive component is the magnetic lock itself which, on average, has a cost of 100€. So, the total costs per room add up to 200€for all the required hardware components.

### *C. Deployment Issues*

PKI being the base of the system, the most crucial component is, of course, the certification authority that will issue certificates for the users. It is recommended that the organization have its own internal CA for issuing certificates. Also, special protection measures must be taken in order to secure the CA private key. An LDAP server is also required for storing the user related information and certificates.

The next component on the list is a database system that will be used to store both the user sessions from the SSCP platform as well as data from the Location, Biometric and SAAS services. The databases and their location must be scaled accordingly to the number of users of the system and the number of protected areas present within the building. Also, since the Biometric service deals with large amounts of image data (a set of 16 images with different lighting conditions is required for each user in order for the algorithm to work correctly), it is recommend that a separate database server be used for this service.

A further requirement is an Apache web server for hosting the FastCGIs for SSCP and the SAAS service. Also, for redundancy and load balancing purposes, a greater number of web servers may be configured. For each secured area, a router will be required. The router will have to have at a minimum two USB ports (one for the Bluetooth dongle and another for the magnetic lock controller) for each door it controls (of course, an USB multiplexer can be added to commercial routers that do not meet this requirement). The router will host The Lock service described earlier and special changes must be made to its firmware in order to be able to host the service. Also, from a hardware point of view, the router must have at least 8 MB of flash memory in order to host the new firmware and the additional modules required for the Lock service.

### *D. Performance and reliability tests*

The most crucial components of the system are the Authentication and Session Check CGI because, in the absence of these two components, any other service that requires user identification would fail. As such, these services must be evaluated for their response time as well as reliability.

Reliability was tested by simulating high request loads, 20 requests/second, on the SSCP Authentication and Session Check CGI as well as the SAAS Key CGI for more than an hour. As expected, the fastCGI module spawned accordingly the necessary number of instances and no failure was detected for those instances during the test.

A detailed breakdown of the medium time spent on an authentication request is presented in Fig. 25. As can be deduced from the graph, the main time consumer is the LDAP check and information retrieval because of the multiple attributes stored by the LDAP directory schema for a single user. SSL context establishment was measured using the Apache log timestamps because the web server is in charge of this operation. For this test, the database, along with the OCSP responder and LDAP directory, was installed on a different physical machine from the Authentication CGI.

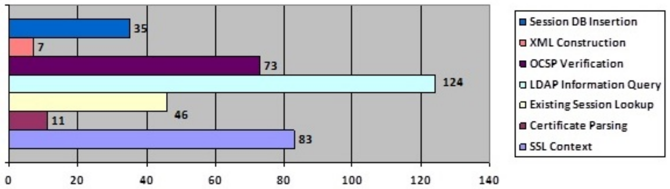

Figure 5. SSCP Authentication Request processing time breakdown.

## VI. Future Work

The proposed system has been designed in order to be able to easily integrate with as many existing access control systems as possible and also provide easy extension. One of the main directions we could follow to expand it is to integrate with RFID cards.

Any discussion of RFID on mobile phones is incomplete without also discussing Near Field Communications (NFC), which is an interface and protocol built on top of RFID and is targeted in particular at consumer electronic devices, providing them with a secure means of communicating without having to exert considerable effort in configuring the network. Future work includes customizing the NFP in order to integrate with our present solution, which would be much better suited, from a security point of view, than what is currently used, namely Bluetooth.

This is an ideal scenario for mobile phones as it would allow them to interact with other devices such as laptops while minimizing battery consumption.

In addition to this, new resources, besides sensitive areas, can be protected by our solution. An example of such a resource is a workstation. An authentication system can be developed and integrated with an already existing operating system. As such, when the user is in front of a computer from within the organization's network, he can use his smartphone in order to get credentials to log into the system, or, if a Bluetooth adapter is present on the workstation, log in automatically when the phone is in range.

## VII. Conclusions

As smartphones become more an more popular due to the advances in technology, so do the needs for more portable and diverse applications increase. These powerful devices can now easily handle computational-intensive tasks such as image recognition, while letting you check your calls and your mail at the same time.

In this paper we have presented a generic platform to control the opening of a magnetic lock using identification through

Public Key Infrastructures. One of the main advantages of this device is that, despite its low cost, it is generic and can also be integrated with other existing services. In addition to this, our approach integrates security measures including identification and localization of the users. Based on this platform we have presented our implementation, as well as several possible test scenarios.

## Acknowledgment

The research regarding the context-aware integrated mobile services is supported by national project: ”TRANSYS - Models and Techniques for Traffic Optimizing in Urban Environments”, Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

CAPIM's official site and source code repository are available at http://cipsm.hpc.pub.ro/capim.

## References

[1] S. Y. C. Hsu and W. Wu. Constructing intelligent home-security system design with combining phone-net and bluetooth mechanism. In *Machine Learning and Cybernetics, 2009 International Conference*, volume 6(1), pages 3316–3323, 2009.

[2] E. Carayannis and E. Turner. *Innovation diffusion and technology acceptance: The case of PKI technology*, volume 26(7). Technovation, 2006.

[3] D. Comaneci and B. Vlad. Face biometric distributed authentication service. Technical report, Faculty of Automatic Control and Computers, Computer Science Department, University Politehnica of Bucharest, 2011. Numeric Systems Architecture Course Project Description.

[4] C. Dobre. Context-aware platform for integrated mobile services. In *International Workshop on Services for Large Scale Distributed Systems*, September 2011.

[5] D. Greceanu. Platform and services for context information agregation and visualizatione. Technical report, University POLITEHNICA of Bucharest, Romania, 2011.

[6] I. Hwang and J. Baek. *Wireless access monitoring and Control System based on Digital Door Lock*, volume 53(4). Consumer Electronics, IEEE Transactions on, 2007.

[7] Y. Y. J. Zhang and M. Lades. Face recognition: eigenface, elastic matching, and neural nets. In *Proceedings of the IEEE*, volume 85(9), pages 1423–1435, 1997.

[8] N. R. P. K. L. Iftode, C. Borcea and P. Zhou. Smart phone: An embedded system for universal interactions. In *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004)*, May 2004.

[9] I. Militaru. Indoor localization service. Technical report, University POLITEHNICA of Bucharest, Romania, 2011.

[10] N. D. N. Ravi, P. Stern and L. Iftode. Accessing ubiquitous services using smart phones. In *Proceedings of the 3rd International Conference on Pervasive Computing and Communications*, 2005.

[11] P. S. Y. Park and J. Pyun. Smart digital door lock for the home automation. In *TENCON 2009 - 2009 IEEE Region 10 Conference*, volume 1(1), pages 1–6, 2009.