Investigating the viability of small scale, easily deployable and extensible hotspot management systems

M. Thinyane, G. Foster and P. Clayton
DEPARTMENT OF COMPUTER SCIENCE
Rhodes University, Grahamstown

e-mail: mthinyane@rucus.net, tel: +27 46 603 8642

Abstract - The proliferation of Public Access Locations (PALs) is fuelling the development of new standards, protocols, services and applications for Wireless Local Area Networks (WLANs). An increasing number of Wireless Internet Service Providers (WISPs) are setting up PALs to provide public wireless network access and in turn also facilitate the growth of ubiquitous networking. PALs are set up at public locations to meet continually changing, multi-service, multi-protocol user requirements. This paper discusses the essential infrastructural requirements that will enable and facilitate further proliferation of PALs. Based on those requirements the paper presents an extensible architectural framework for PAL management systems that inherently facilitates provisioning of multiple services and multiple protocols on PALs. The framework that we propose takes into consideration and supports the implementation of diverse business models that facilitate next-generation network service usage accounting. The paper then presents a system that implements the framework and discusses the tests that were performed on the system to validate its effectiveness in meeting the requirements for small-scale PALs.

CATEGORIES: Access Network Technologies, WiFi - 802.11, Hotspots

I. INTRODUCTION

The rise in the proliferation of wireless computing [1] has been facilitated by the developments in wireless communication technologies and also by the availability of mobile devices that are enabled to operate on wireless networks. The move towards ubiquitous computing has been driven by the need on the part of the consumers to always be connected, and to always be online [2]. As a result there's been a steady increase in the number of Wireless Internet Service Providers (WISPs). These WISPs are responsible for providing wireless Internet connectivity to consumers. A large potion of WISPs are companies that provide wireless internet service as one of their core business lines e.g. Telkom T-Zone, MWEB, T-Mobile. The business model that these WISPs implement is the retail model where they sell the service directly to the consumers; these WISPs are also referred to as plain WISPs [1]. The other WISP business model that has been on the rise is the wholesale WISP model, where the remote access companies and mobile carriers own and maintain the hotspot infrastructure and provide access to other businesses who in turn provide the service to the end users [3]. This model has seen a lot of partnerships being formed between businesses i.e. restaurants, coffee shops, bookshops and the service providers, examples of these partnerships include McDonalds and Wayport, Mweb and Mugg 'n Bean [4], [5]. In this model, the businesses benefit from the technical expertise and support provided by the service providers and the service providers in turn benefit from the wide coverage and the established client base of these small businesses.

This paper focuses on single point WISPs [1]: These are small businesses whose first line of business is not wireless service provisioning and who may provide the wireless connectivity to their customers as a complementary service. In the case of these businesses there's no technical expertise provided by a wholesaler WISP and the management of the Public Access Location (PAL) is exclusively the responsibility of the business. The paper identifies the PAL infrastructural requirements of these small businesses and subsequently proposes an extensible, easily deployable framework for small PAL management systems. The later sections of the paper detail an actual implementation of the framework on a system called SEHS (Small Extensible Hotspot System).

II. REQUIREMENTS FOR SMALL PALS

The requirements of PALs differ depending on the specific environment in which the PAL is set up. The main differentiating factors between hotspots are the physical size of the location, the number of simultaneous users and the types of usage expected on the hotspot [6]. In this paper, a small PAL is taken to mean one that exhibits the following features, most of which are defined in a study done by Intel [6]:

- The maximum user density is 10 users per Access Point (AP).
- The horizontal to vertical coverage is 1500 sq ft to 6 ft respectively i.e. within coverage of a single AP.
- Simple user authentication with a minimum level of encryption is the level of adequate security that is required on these PALs.
- Most of these small PALs are single proprietorship businesses.

- These PALs have access to limited technical expertise
- Most of these PALs have a minimal computing infrastructure in place with internet access.

We hereafter provide an evaluation of the specific requirements of small PALs. These requirements have been determined both from literature survey and by questionnaires distributed among small businesses that could potentially provide wireless service. The main requirements were ascertained as follows:

- Extensibility: the overarching requirement for small PALs is the extensibility of the underlying management system in terms of being able to implement extra functionality. This is due to the fact that different WISPs have different characteristics that determine the kind of service that is provided on the PAL. These differing characteristics influence the business model implemented, the charging model implemented, the customer-vendor relationships that are established and the roaming relationships between partnering PALs. The proliferation of these small WISPs also facilitates the development of new and different service element modules and so the management system should be able to implement these different services on the PAL.
- Authentication, Authorization and Accounting (AAA) considerations: the PAL has to take into consideration the functionality related to authenticating users on the network, authorizing their use and accounting for the network usage. Authenticating users on the PAL is important in the light of need to account for network usage. Authorizing users becomes an important feature where different users should have different privileges to network resources. Accounting provides access to network usage information.
- Usability: this is a feature of the system that determines how easy it is to use the management system. While usability is largely a subjective characteristic, it is however strongly influenced by the simplicity of use of the system. Therefore usability coupled with simplicity also becomes a major requirement for these PALs.

III. XOBOGEL: THE ARCHITECTURAL FRAMEWORK

We have developed the Xobogel framework to provide an architecture that addresses the above mentioned requirements. The underlying design pattern that forms the architectural basis for Xobogel is the microkernel pattern and it is interlaced with the Internet Protocol Detail Record (IPDR) reference model [7]. The framework is designed to take advantage of and to exploit the best features of these existing technologies and protocols.

The microkernel pattern provides the basic infrastructure that facilitates extensibility, and in turn service differentiation, in the framework [11]. The IPDR specification, on the other hand, provides a means for the exchange of usage data between different system units and allows for the integration of the implemented system with the WISP's Business Support System (BSS) [7]. The IPDR model uses a layered architecture to define the collaboration of different modules and components in the system [Fig. 1].

The network and service elements layer comprises all the service modules and elements that provide the network service to the clients. The mediation layer provides an interface between the BSS and Network Element layers. It is responsible for the collection of usage data, encoding the data into IPDR format and transmitting the IPDR document to the BSS system. The BSS layer implements the business operations that are relevant to the provisioning of the network service e.g. billing, data mining, and network usage analysis.

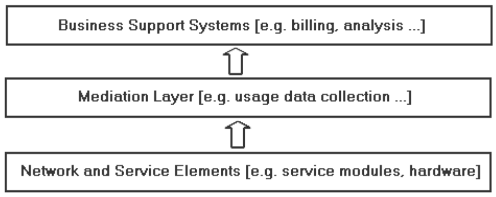

Fig. 1. IPDR layered architecture, adapted from [7]

A. Xobogel High Level Model

Xobogel high level architectural model encompasses elements from the IPDR specification and the microkernel pattern to define a model that meets the requirements identified as necessary for multi-service, easily deployable and extensible PAL management systems [Fig. 2].

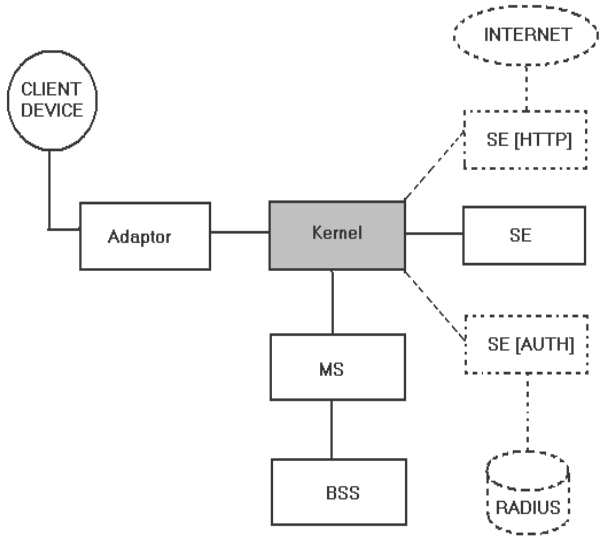

Fig. 2. Xobogel high level model

The different components of the model collaborate as follows:

- Kernel: is the central, core component of the model that implements minimal functionality to facilitate communication between the different components of the model. It serves as a socket for plugging in service element modules into the system.
- Service Element (SE): equates to an internal server in the microkernel pattern and it provides the functionality needed to implement the network

service. The service elements plug into the core Kernel. Extending the functionality of the system essentially entails developing a new SE module and plugging it into the Kernel (e.g. security-related features; data integrity, non-repudiation, confidentiality, etc). Two samples of SEs are shown in [Fig. 2], Authentication SE and HTTP SE.

- Mediation System (MS): implements the functionality needed to allow the BSS to operate on the information from the SE. It collects the raw usage information from the SEs and formats it for the BSS.
- Business Support System (BSS): implements the business level operations that are necessary for the provision of the network service.
- Adaptor: provides a connection between the clients and the system functionality. It provides an interface into the system via which the clients can request services and receive responses.

The Xobogel framework implements a plug-in architecture that allows the different service elements to be plugged into the kernel. Once a service element module has been loaded, the client devices are then able to request the service via an adaptor [Fig. 3] that provides the communication channel to the service elements.

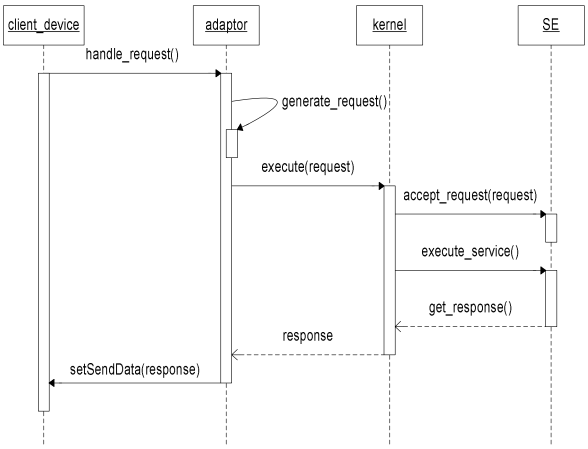

Fig. 3. Service Element usage sequence diagram

In order to facilitate network usage accounting for the PALs, each SE module defines the metrics associated with its usage. This is done via an XML schema definition file which is used by the mediation system to format the usage data into IPDR Document format [Listing 1].

Once the data has been formatted [Listing 2], it is then accessible to the BSS or any other business units for further processing. This allows every SE to encapsulate the functionality to execute the required service and also the information on how to account for its usage.

The Xobogel framework provides an architectural template and pattern for developing management systems for small PALs. The framework has been designed to be abstract and general enough to be implemented in different environments.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
     targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
     version = "3.0-A.0"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">
    <include schemaLocation = "IPDRDoc3.0.xsd"/>
    <element name = "clientIP" type="string"/>
    <element name = "startTime" type = "dateTime"/>
    <element name = "endTime" type = "dateTime"/>
    <element name = "usageUnit">
        <simpleType>
           <restriction base="string">
              <enumeration value="timeSecs"/>
              <enumeration value="timeMins"/>
              <enumeration value="dataBytes"/>
              <enumeration value="dataKbs"/>
           </restriction>
        </simpleType>
    </element>
    <element name = "quantity" type="int"/>
    <complexType name = "X_HTTP">
        <complexContent>
            <extension base = "ipdr:IPDRType">
                <sequence>
                    <element ref = "ipdr:clientIP"/>
                    <element ref = "ipdr:startTime"/>
                    <element ref = "ipdr:endTime"/>
                    <element ref = "ipdr:usageUnit"/>
                    <element ref = "ipdr:quantity"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

Listing 1. A sample XSD file

```
  <?xml version="1.0" ?>
- <IPDRDoc xmlns="http://www.ipdr.org./namespaces/ipdr" xmlns:xsi="htt
    xsi:schemaLocation="http://www.ipdr.org./namespaces/ipdr x_http.x
    0a9bc0a80064" creationTime="2005-01-22T08:38:41.392Z" IPDRRecord
  - <IPDR xsi:type="X_HTTP">
      <IPDRCreationTime>2005-01-22T08:38:41.392Z</IPDRCreationTime>
      <seqNum>1</seqNum>
      <clientIP>192.168.0.101</clientIP>
      <startTime>-1718481985</startTime>
      <endTime>-1718481438</endTime>
      <usageUnit>dataBytes</usageUnit>
      <quantity>2664</quantity>
    </IPDR>
  - <IPDR xsi:type="X_HTTP">
      <IPDRCreationTime>2005-01-22T08:38:41.392Z</IPDRCreationTime>
      <seqNum>2</seqNum>
      <clientIP>192.168.0.101</clientIP>
      <startTime>-1718481188</startTime>
      <endTime>-1718481157</endTime>
      <usageUnit>dataBytes</usageUnit>
      <quantity>3323</quantity>
    </IPDR>
```

Listing 2. Formatted usage information

IV. SMALL EXTENSIBLE HOTSPOT SYSTEM (SEHS)

The study that was undertaken to determine the characteristics of small PALs provided a set of features that guided the implementation of the Xobogel framework in a system called SEHS. These guiding factors are as follows:

- A large portion of the small businesses have a computer infrastructure in place, and the common operating system running on these computers is MS Windows. As a result SEHS was developed specifically for the MS Windows platform. The language of choice for implementation is JAVA and this is to take advantage of the Write Once Run Anywhere (WORA) feature of JAVA [8].
- The billing model preferred by these businesses is predominantly usage-based billing. Preferences for flat-rate and free-usage models were varied. This directly influenced the sample billing modules that we implemented in SEHS.

- Most businesses would prefer to charge for network usage immediately as opposed to charging on account. This means SEHS has to cater for pay-as-you-go billing method (via coupons) and also account payment method for customers who have a formal relationship with the business.
- A users' questionnaire indicated that the top uses on PALs are e-mailing and web browsing. This directly influences the service element modules that are implemented on SEHS.

The underlying design guideline for SEHS is to provide an extensible architecture and this has been archived via the use of JAVA language Object Reflection features [9]. A JAVA interface is defined for all the pluggable components of the system i.e. x_serviceElement [Fig 4] for the service element modules and x_billModule [Fig 5] for billing modules.

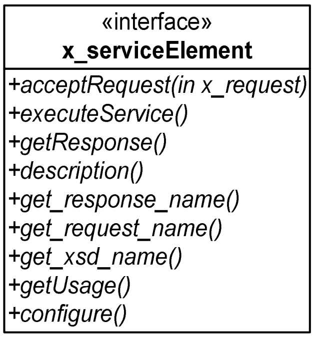

Fig 4. x-serviceElement interface class object

```
import java.util.*;

public interface x_billModule
{public void configure(String se_xsdname);
 public String description();
 public void charge(ArrayList usage);
 public void reset();
 public String formula();
}
```

Fig 5. x_billModule interface definition

Extending the functionality of the system simply entails loading a JAVA class object that implements the specific interface into a module directory and specifying the associated schema definition file. The system then determines the modules that are loaded, using the JAVA reflection API [9], and makes these available for use in the system [Fig 6].

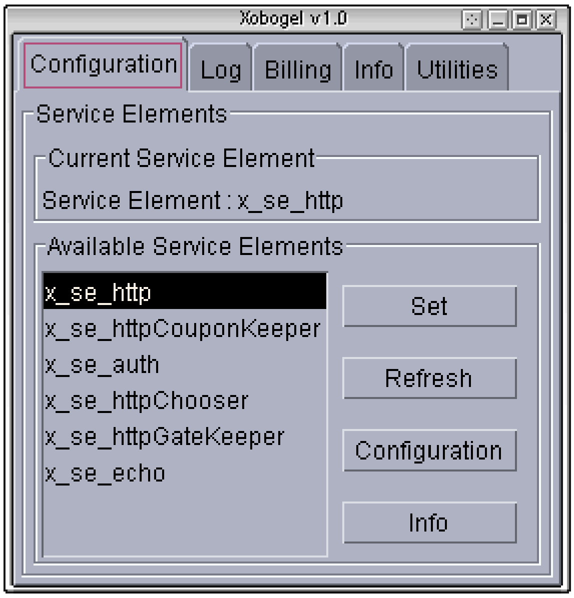

Fig. 6. Available service element modules

A. Usage information semantic ambiguity

The nature of SEHS is such that it allows for different modules, from different developers to be implemented and loaded into the system (i.e. different billing modules have to operate with different service element modules). This invariably creates a problem of semantic ambiguity as far as the usage metrics per service element module are concerned.

Semantic ambiguity occurs in situations where a meaning of a word must be determined with the help of greater knowledge sources [10]. Some of the methods that are used in language studies for sentence disambiguation are:

- Constraining the lexicon: this is limiting the words that are used in the language only to those that do not carry any ambiguity. In SEHS, this would imply reducing the number of metrics that can be used for usage accounting on each service element to a set of common attributes or tokens.
- Constraining the source language construction: in language studies this is done by defining a grammar for the language: a set of rules that define sentence construction.
- Interactive disambiguation: this is used where automatic disambiguation is not possible. It involves interacting with human consultation.

Interactive disambiguation is the method that is used in SEHS to facilitate the interoperability of different billing modules and service element modules. In SEHS this is achieved via a process of dynamic interactive attribute mapping [Fig. 7] between billing module variables and service element usage attributes. Once a billing module has been loaded, it queries the underlying service element module to expose the associated usage attributes, and then allows the user to map the corresponding billing module variables [Fig. 8].

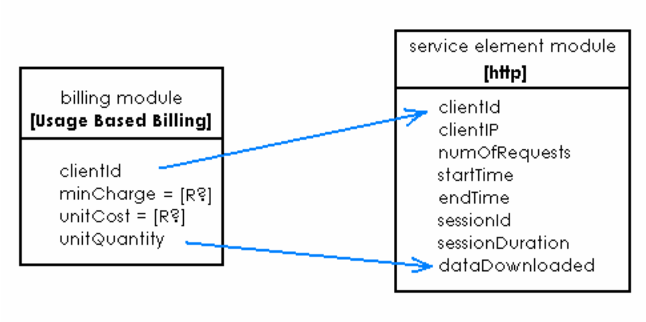

Fig. 7. Attribute mapping illustrated

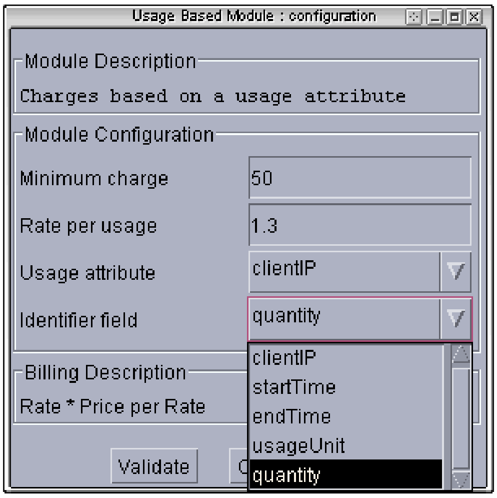

Fig. 8. Billing module variables mapping

This allows the billing modules to be changed and reconfigured to work with the loaded service element module without any need to either shut down the system or recompile the system source code.

B. Stress testing SEHS

In order to determine the robustness of the implemented system and for the purpose of observing how the system behaves under increasing user loads, we setup a stress testing session on SEHS. The experiment simply involved simulating multiple requests for a local (1 MB) file. Since the target PALs have an assumed density of 10 users per AP, the number of simulated users in the experiment was limited to 20. This is also due to the limitation in the edition of the stress testing tool[1] that was used. The experiments were executed from a localhost and therefore show the potential performance of SEHS without taking into consideration the limitations in speed associated with 802.11.

The trend that is worth noting is the steady increase in the time it takes to complete a web session as the number of users increases [Fig 11]. There's a linear relationship[2] between the number of users and the average session time. [Fig 10] shows the average time it takes to establish a connection with SEHS. Since no connections were recycled in this experiment, there's also a general increase in the time it takes to establish a connection as more user get onto the network despite the fluctuations at 15 users. There is a sharp decrease in the throughput observed on the network as users increase followed by a steady decline from 10 users. This is observed in [Fig 9], which shows the average network throughput of the fastest URL, which has exchanged at least 4096 bytes of data.

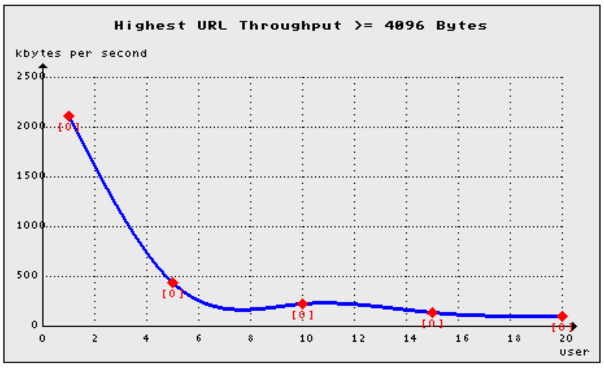

Fig. 9. Highest URL throughput

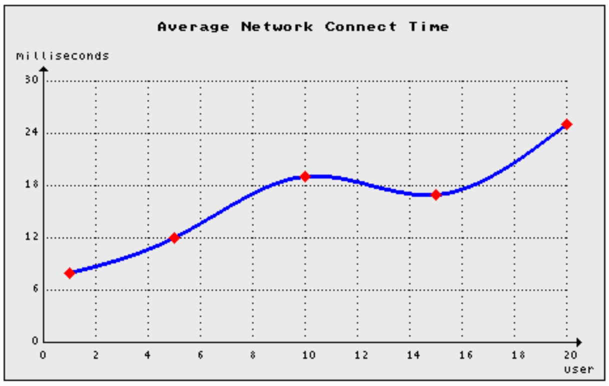

Fig. 10. Average network connection time

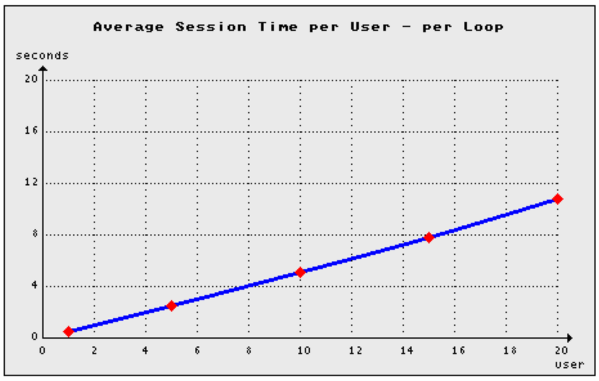

Fig. 11. Average Session Time per user, per loop

V. Benefits and Conclusion

The Xobogel architecture combines the flexibility and adaptability of the microkernel pattern with the portability of the IPDR document format [7]. The framework provides the following benefits;

- *Reliability*: The separation of the SEs from the Kernel achieves a more robust architecture that

---

[1] The tool that was used in executing this experiment is Proxy Sniffer™, available at http://www.proxy-sniffer.com

[2] The trend line equation for Fig 11 was determined using MS Excel as y = 0.5389x - 0.1385 and the correlation coefficient was computed as 0.999382

isolates SE related problems from affecting the whole system. The stress test results [Fig. 11] indicate a linear relationship between the number of users on the system and the average session time, which also attests to the system's reliability in terms of how it scales.

- *Portability*: the architecture separates the service-specific implementation factors from the high level system management considerations. The kernel is separated from the SEs by a level of abstraction that defines an interface that should be implemented by the SE.
- *Modularity*: the architecture comprises components that are modular and that encapsulate the system functionality.
- *Flexibility*: the decoupled architecture allows for flexibility in terms of changing and modifying the components of the system without having to alter the whole system. This flexibility is owed in part also to the microkernel pattern that the framework is based on [11].
- *Integration* into WISPs' BSS is easily achieved. This is due to the fact that the architecture provides the network usage data in an XML based IPDR Document format which allows WISPs to use it as fits their business model. The system can also be easily integrated into any IPDR compliant system.
- *Ease of implementation*: The functionality related to the IPDR specification is also easy to implement due to the free availability of open IPDR libraries. The framework's plug-in architecture allows for the system to be implemented using already developed components (e.g. service elements, billing modules).

The architectural framework successfully meets the requirements that were identified for small PALs. The implementation of the framework in SEHS and the resultant experiments that were performed on the system indicate the applicability of the framework in providing an extensible, easy to use PAL management system.

ACKNOWLEDGMENT

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom, Business Connexion, Converse, Verso Technologies, THRIP and the Andrew Mellon Foundation

REFERENCES

[1] Alven D., Farhang R. (2002, February), *"Does it take a WISP to manage a wisp of hotspots?"*. A Masters Thesis at IMIT - Royal Institute of Technology, Stockholm, Available [Online] http://web.hhs.se/cic/courses/underthebridge/wisp_analysis.pdf

[2] Minoli D., "*Hotspot Networks: Wifi for Public Access Locations*", USA: McGraw Hill Networking, 2002, ch 1.

[3] Probe Research Inc., "*Joltage shocks the PWLAN Market*", [Online] Available: http://www.lightreading.com/document.asp?doc_id=28748&print=true

[4] Fleishman G., (2003, June) "*McDonalds to launch in three cities with three providers*", Available [Online] http://www.wifinetnews.com/archives/001839.html

[5] MWEb, "*MWEB Wifi Access (South Africa)*". Available [Online] http://www.mywifi.co.za/providers/mweb_wifi.htm

[6] Hammond J. (2003, December), "*Wireless Hotspot Deployment Guide*", Whitepaper - Intel, [Online]. Available: http://www.nor-tech.com/solutions/dox/Wi-Fi_Hotspot_Deployment_Guide.pdf

[7] IPDR.org, (2004) *Internet Protocol Detail Record*. [Online] Available: http://www.ipdr.org

[8] Curtin M., (1998), "*Write Once, Run Anywhere: why it matters*", Sun Microsystems Inc., [Online] Available: http://java.sun.com/features/1998/01/wora.html

[9] Green D., "*Trail: the Reflection API*", [Online] Available: http://java.sun.com/docs/books/tutorial/reflect/

[10] Baker L. K., Franz M. A., Jordarn P. W., *"Coping with ambiguity in knowledge based natural language analysis"*, Proc of COLING-94, pp 90-94, Available [Online] http://www.lti.cs.cmu.edu/Research/Kant/PDF/flairs.pdf

[11] Buschmann F., Meunier R., Rohnert H., Sommerlad P and Stal M. (1996, August), *"Pattern-Oriented Software Architecture: A system of patterns"*, Volume 1, John Wiley & Sons.

BIOGRAPHY

Mr M. Thinyane is currently reading for a final year MSc in Computer Science at Rhodes University, Grahamstown. He completed both his BSc (Information System) and BSc (Hons) at Rhodes University. His research interest is in the area of wireless networking.