



## XAP - AN OPEN SOURCE PROTOCOL FOR eHEALTH

A.F. van der Merwe<sup>1</sup>

<sup>1</sup>Department of Industrial Engineering  
University of Stellenbosch, South Africa  
[andrevdm@sun.ac.za](mailto:andrevdm@sun.ac.za)

### ABSTRACT

In a recent SAIIE conference paper it was suggested to use xAP protocol in the telemedicine environment. Telemedicine has since been described to evolve into eHealth where the focus is not only on the remote control of health systems, but now also includes local control. This wider scope is necessary to allow seamless communication amongst systems, irrespective to whether they are in the same room or in a remote location.

Local communication is via intranet with a higher data transport rate than internet. Larger data packages can be transported than economical on the internet. This research proposes a framework using a xAP datagram instead of sending the complete data package. It will only broadcast the availability of a data package to both local and remote systems. The bulk content can then be requested for download to local and remote systems either in compressed or uncompressed format. The xAP datagram carries information on the formats that the bulk data can be downloaded in. Intelligence at the retriever would decide which format to download. The compression format can be negotiated at download time. Protocols utilized are UDP for xAP broadcasting and TCP for bulk data retrieval.

Scope: It is important to note that this research is on the efficacy of a datagram to broadcast data availability announcements and not on the design of a database. Also that data retrieval via TCP is not considered as it is used for bulk content retrieval after the announcement and only when an internet link is available. Lastly the bulk content download format negotiation is already available by others, usually "APP" developers of smart devices or similar.



## 1 INTRODUCTION

Many manufacturers of medical devices hold their well researched patented communication protocols close to their chests. However, in South Africa as in the rest of the world, peoples' lives are priceless. There is no compromise or no cost too high to save someone's life. An opensource digital communication language or common protocol is required for all to share patient data without restriction.

Simple information gathered on a patients health condition only needs a "flag" that the patient has been tested. Such a *basic information pack* could include the patients identification, the date and time, the test type and the result in abbreviated format. More detailed analyses, prognoses and doctors notes could follow on request. This means that as soon as a patient has been tested, the test and an abbreviated result should be broadcast to all interested systems or devices. Secure information can be held encrypted until released by the patients authorization. The existance and accessability of patients information is normally more important than the security thereof.

The success of the Internet lies in its pull strategy. A sentralized eHealth database server would retrieve the patients history, details, prognoses and medications. However, when the internet is off, or where out of Global System for Mobile Communications (GSM) reach, like in rural Africa, another strategy is required. Ferrer-Roca et al (2004) suggested a systematic management of diabetic patients using now basic sms messages.

A push pull strategy is recommended under these circumstances . The broadcast (push) of a *basic information pack* could trigger whenever an event occurs. An event could entail any interaction of any patient with any medical service. More detailed data could then be retrieved using a secure https pull system as necessary and when internet becomes available.

The combination of the push of events as they occur by xAP and pull of more information by remote doctors for prognoses or of the central robots for data assurance naturally reduces data traffic. A reduction in data traffic is what is required by telemedicine which has to operate into areas where internet is intermittent and often offline. The proposed xAP mechanisms will function equally efficient over dialup connections whether GSM or landline.

In this paper the concept of data transmission in ehealth investigated, device types defined and a look at the file types that would be required. Finally the xAP schema and implementation is concluded.

## 2 DATA TRANSMISSION IN EHEALTH

In eHealth there is a rapidly developing amount of clinical information available on each patient. The main aim of this study is not to broadcast the electronic information of each patient, but to broadcast the index of where a patient's information is kept. Duplication of information even to be admitted to a hospital is becoming an administrative challenge. The internet is providing a powerful link between client and server. With modern systems, clients can also be servers, and dedicated high performance servers are not required. Storage space on basic computers can be several terabytes, and on handheld devices several gigabytes. The real challenge is to transmit relevant patient information quickly and efficiently in such a manner as to "follow" the patient. Later in this paper it is considered that patient global positioning coordinates from his smartphone can be used to locate a him in an emergency, and even advise him of the nearest medical center.

We now consider some advances in technology and the way devices would communicate:

## 2.1 Data types

Several data types are required to store and transport data within the telemedicine system. These were identified by Van der Merwe (2010) as patient identification data, test operator data, device identification, test identification, test device calibration, test result summary and test result detail. Test results would typically require more data than patient identification data. The system therefore has to determine what data to broadcast and what data to hold, ready to transmit on request. Factors that influence this decision are the file type, the storage capacity at the requesting device and the transmission link between the transmitting and requesting systems. Local devices on LAN would share greater volumes of data than remote devices on GSM modem or SMS. A negotiation protocol with fallback has to be used. Several such systems are already available where internet based websites intelligently determine the type of device requesting the information as well as the data connection speed and quality. The website server would then reformat the data transmitted to best suite the requesting device. An example of this is the browsing of a website which has been enabled for mobile access. The website server identifies the requesting device as a mobile smartphone and “simplifies” the webpage to a format and content best suited for the phone display. The user gets the benefit of a simplified interface which improves his accessibility of the information required. Several banking institutions have implemented this in their mobile banking services. Varshney 2006 suggested the use of wireless local area networks (WLAN) and personal area networks (PAN) to communicate with lan based devices.

Another smartphone development requires the download of a small application to the smartphone. The application is optimized for that type of smartphone, and provides an interface to web based data in a user friendly format. Many such applications already exist for all smartphones on the market. The advantage is that only basic information has to be transmitted, and the interface speed is now limited by the speed of the smart device rather than by the datalink speed. Flexibility is more restricted, but is largely overcome by regular updates, which in itself is often hampered by backwards compatibility requirements.

## 2.2 File types

Patient personal information can be transmitted in text well within the limiting 256 ASCII characters available. High resolution colour imagery required in prognoses to assess skin condition could require high data capacity. Capturing is done at high resolution and transmitted at lower resolution, based on the size agreed on between the server and client. Smart applications found on modern tablet like iPad, Playbook and others can better negotiate the resolution.

## 2.3 Application Layers

UDP is a transport layer protocol. xAP is a UDP message formatted as a text string, which conforms to a specific schema. Smartphone applications communicate on an application layer typically using UDP or TCP to transport messages. The use of xAP the format is therefore a way of formatting the UDP messages sent between smart devices. In smart devices we would include all smartphones, tablets, personal computers and any internet layer connected device capable of sending and receiving UDP data. Van der Merwe (2010) previously discussed the general usability of LAN, WLAN, TCP, UDP, GSM, GPRS/3G, SMS, MMS, and RF. Taluker and Das (2010) presented the architecture and technology suitable for mobile web for the under-privileged. Mobile web will bridge the gap between have and have-nots and will help eradicate digital divide. they presented how SMS can be used as a transport bearer for mobile web. This is suitable for xAP protocol transport.

As part of this study, several tests were conducted using social network application layer interfaces. Twitter was tested using tweets set up according to the xAP schema. Albeit a delay due to the Twitter update speed, the message was received, parsed and executed by the smart device without error. Reliability of such social networks could cause concern to those relying on life threatening information. However, during this experimental phase all avenues are considered, and can be classed according to security and reliability when required. Such commonly available networks freely available and are becoming more stable due to consumer demand.

## **2.4 The Proposed xAP framework**

The xAP framework for eHealth proposed here is part of ongoing research. Several of the building blocks have already been tested successfully. However, significant in field usability testing is still to be done. The following describes the framework components to best fit this application:

### **2.4.1 Device types**

Devices can be classified as smart devices and non-smart devices. Several categories of smart devices are foreseen, but initially a smart device should be defined as a device that can communicate using UDP protocol using xAP text format. A non-smart device would require a translating interface device to enable it to communicate using xAP. The translating interface device would run a software driver that translates the non-smart device's output to a xAP format and transmit it over the network. Controlling commands sent to the non-smart device would control available functions.

### **2.4.2 Local and remote devices**

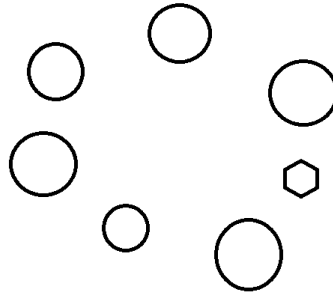
Local devices are connected, monitored and controlled at a single location. Often these devices are non-smart and require a computer or other network interface (barionet). Typically connection types are RS232, RS485, USB, Bluetooth, ANT, RF, computer mouse & keyboard, movement sensors, movement camera, other short distance wireless and temperature sensors. LAN devices can be classed as local devices. More modern sophisticated devices can connect to a network wired or wireless. These include media streaming players, tablets, human interface devices, movement cameras.

Where data transmission over public internet is required like these are classed as WAN devices. These devices are normally accessed through static IP internet GPRS & 3G. For example the GSM Commander is a simple logic PLC which can be configured and monitored via GPRS. Another example is an access control system called gate keeper. Human interface devices like tablets & smartphones switch seamlessly between WiFi and 3G.

Testing UDP over Vodacom 3G network required a special application for an unrestricted apn at every device. The 3G router has to run DynDNS. Each device would then connect to the URL of the router and broadcast its UDP to that URL.

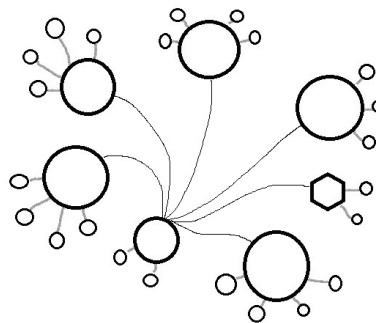
### **2.4.3 Device configurations**

In figure 1 several smart devices are shown as large circles. These could include database computers or basic temperature logging systems. The network itself is not shown in the diagram. The hexagonal device in this example could represent a mobile hospital or emergency unit such as an ambulance.



**Figure 1: Groups of smart devices connected to a network**

In figure 2 several non-smart devices are shown connected to the smart devices. Smart devices can also be used without interfacing with non-smart devices. The lines from one smart device illustrate a broadcast of a UDP message to the other smart devices. Each other smart device would listen if the message concerns them or not. The protocol provides for wildcard messages that can be send to groups or types of devices.



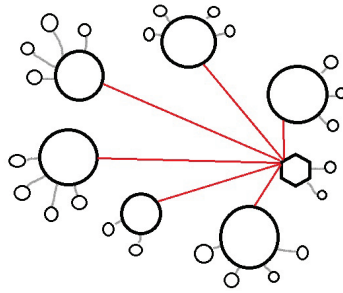
**Figure 2: Smart devices with -non-smart devices during a broadcast**

#### **2.4.4 Device "drivers"**

Device drivers are programs running on the smart devices. These programs communicate to connected non-smart devices through fixed or wireless interfaces. These interfaces include Bluetooth, radio frequency, ANT, relay switches, logical inputs, RS232, USB, RS485/422, analogue, pulse width logic. The driver translates the input from the non-smart device to a xAP format UDP message and transmits to the network. The driver also test communication to the non-smart device and transmit a heartbeat to the network on behalf of the non-smart device. The driver will react on queries put to the non-smart device. Drivers are designed to execute in multitask fashion on the smart device.

#### **2.4.5 Data cache servers**

Every device, whether smart or non-smart, has to be connected to a smart device capable of caching short term historic data. These smart devices are termed cache servers. When data transmission is lost between devices, a listening device may request information by means of a query. The server would then send the latest information on the status of the queried device parameter. The server can be programmed that if it receives several queries, to recognize that it may have been offline. A broadcast burst of recent historic data may then be triggered. Figure 3 shows diagrammatically a query from the mobile unit when it becomes online after having been outside of network reach. In response to the query, each queried device would send updated its data.



**Figure 3: A xAP query is generated by the mobile unit after being offline**

#### **2.4.6 Human interface**

Our modern challenge is to sort and make sense of the abundance of information we receive. The human still only has several senses, of which the eyes are most commonly used in data communication.

The eyes can capture a large picture where the ears require more time to acquire the same amount of information. The ears however function even when the eyes are not, and can be utilized to receive information over a prolonged period. For example while driving, an audio signal can be received, but watching a visual display is not desirable. When a doctor is operating, his commands are in audio, as his eyes and hands are occupied. Audio human - machine interface is already used extensively to warn the user of changing machine status. Commanding of the machine on the other hand is still a topic of ongoing research as generic speech recognition is not proven to be reliable and is often not preferred.

Haptic control and feedback is becoming more important as a sensory interaction with the human user. Vibrating cellphones alert the user of incoming communication whilst not disturbing his visual or audio interaction. Such incoming communication can then be attended to during the next pause in the current visual task.

Video games use vibration feedback to communicate with the player. We sense vibration or change in noise type when a fault in our cars occur. Using inexpensive piezo electric devices, both sensory and actuary information can be interfaced with the human user.

Human senses of smell and taste are also being explored. Experienced medical personnel could tell by smell if an adverse condition exists with a patient. A "smelling" device connected via ANT technology to a patient's smartphone could send condition information to the eHealth system. Such sensors would have to be developed. The iPhone can be already connected to ANT technology. ANT is a low power radio frequency protocol used by Garmin to communicate with their heart rate monitors.

The raise of an alarm with the patient, and the eHealth system would require a simple application running on the phone. Existing GPS location information like Google Latitude could be added to the patient's condition upload. The medical center closest to the alarmed patient would receive a pre-alert, and the patient can be given directions (part of Google Latitude on Google Maps) to the medical center. Should the system not receive regular updates, the registered patient can be alerted via sms or voicemail. This technology would give a patient the freedom of a normal life with the comfort of medical support.

Both medical personnel and patient will benefit from such improved communication.



### 2.4.7 Text schema

The xAP schema specification was developed by several specialists mentioned on [www.xapautomation.org](http://www.xapautomation.org). The specification is widely used to carry automation triggers over UDP/IP between UDP enabled devices. The xAP protocol is solely text based, which makes it simple to program and debug. However, text only data has its limitations. Van der Merwe (2010) discussed more detail on the theory of the text message format.

In WAN tests using Twitter as a communication carrier with the following typical message requested an alarm status from a smart device. The smart device is a Hometroller home automation system with embedded XP operating system connected to a non-smart device. The non-smart device is a CADDX NX5e alarm system communicating through RS232 only. The Hometroller runs a plugin (device driver) which translates the alarm system status to a device status. On device status change another plugin transmits via Twitter. The Twittseer plugin also listens for tweets, and is programmed to react on recognized xAP queries like the following:

```
xap-header
{
v=13
hop=1
uid=FF.000E:5E04
class=xAPBSC.Query
source=mcs.Xap.SAM:Caddx.Panel_Status.5E_4.Panel_Status
}
input.state
{
ID=*
}
```

The response is a xapbsc.info message parameters:

```
xap-header
{
v=13
hop=1
uid=FF.000E:5E04
class=xAPBSC.info
source=mcs.Xap.SAM:Caddx.Panel_Status.5E_4.Panel_Status
}
input.state
{
Status=disarmed
}
```

A command to set the alarm would be:

```
xap-header
{
v=13
hop=1
uid=FF.000E:5E04
class=xAPBSC.cmd
source=mcs.Xap.SAM:Caddx.Panel_Status.5E_4.Panel_Status
}

state
```





```
{
Status=arm away
}
```

### 3 IMPLEMENTATION

Not all eventualities can be simulated in this experimental phase. This part was published in a previous conference proceeding and has now been improved. An attempt is made to propose an architecture which would overcome the basic requirements of a push pull system in eHealth and mHealth. The testing of xAP communication between several LAN based devices and various personal has now been continuous for more than two years. Reported failures were limited to when devices or computers stopped working. No occurrences of lost data could be found. On networks more complex than LAN error basic checking should be considered. Due to the broadcast nature of the protocol, such error checking can be done using the protocol's heartbeat. If a heartbeat has not been received by a listening device, a query can be sent to the sleeping device. Sleeping devices can be reported to a systems administrator robot to take corrective action. Administrator robots can normally use TCP control to check and correct sleeping devices. A fallback would be a simple telephone call to the site of the sleeping device to request a power cycle of the device.

#### 3.1 Sequential Heartbeats

Although not part of the standard protocol, a sequential heartbeat is used. A timestamp string is used which contains the time value (similar to that used in Excel) related to the time of transmission. The listening device checks for regular intervals between received heartbeats and flags non regular occurrence. System service level can be monitored on this information. The xAP text message would look like this:

```
xap-hbeat
{
v=13
hop=1
uid=TM.0002:0001
class=xap-hbeat.alive
source=clinic0002.ECG0001
interval=60
port=52000
PID=4556
Timestamp= 40735.4568613426
}
```

#### 3.2 Protocol format examples:

The transmission of xAP messages was successfully tested amongst several personal computers running Visual basic in Excel. The transfer of typical patient data was simulated. In device to database transmission, reliable data transmission was found using twenty five Barionet ([www.barix.com](http://www.barix.com)) programmable logic controllers. Robotic control using haptic interfaces was tested as part of telemetric operations with haptic feedback. All devices were connected to a LAN.

Below we look at two examples of xAP message format we used to transfer data in the telemedicine network. The first is on the entry of a new patient's information, and the second on a simulated ECG device report:





### 3.2.1 Example 1: New patient database entry

Firstly each header of each message should be unique to each patient. A patient's national identification number is used. Where the person does not have a national ID, a sequential number is generated from a common list. The purpose of this message is to introduce a new patient to the network, or to change information on an existing patient. The event would be classed such to change the contact information of the patient. The header and message area would then look like this:

```
xap-header
{
v=13
hop=1
uid=TM.0002:0000
class=xAPBSC.event
source=clinic0002.7502185028080.infodb.Florence
}
database.name
{
FirstName=John
Surname=Smith
}
database.contact
{
Cell=0821313382
}
Database.history
{
ActivationDate=20100601
ChangeDate=20110703
GPS=51.28.38,18.12.01
}
```

Note the UID firstly identifies telemedicine (TM) then the clinic where the patient was introduced as clinic 0002 then the system parameter which in this case is the basic info parameter "0000". The class tells us that it is an event of the xAPBSC type. The source again identifies clinic0002, and the ID number of the patient, followed by the type of content that the message contains namely database info, and the operator who was logged into the system at the time of transmission who was Florence. A receiving device that is not concerned with the "infodb" of a patient would ignore this message. However, every clinic would have at least one system updating the patient's data which would then store this info in its files.

Due to the fact that nurse Florence entered (or changed) three types of database data, the message area consists of three message blocks. In this case the first message block contains the name information FirstName and Surname and the person who made the last change. Similar for blocks 2 and 3. It can therefore be seen that this system would broadcast only in the event of change of information and send only the information that changed. The activation date would be sent to enable the receiving system to compare with today's date to determine if this is a new patient or not.

The enrollment of a patient with limited information can therefore be done by any clinic and the file, although containing only partial information will be broadcast to all other clinics when online. There is no need for a central database for enrolment and for all



clinics to be online all the time. This mechanism enables mobile clinics to venture into Africa and enroll new patients which will be automatically uploaded when an internet link becomes available. GPS coordinates of where the patient was last treated should be included in the patient's "database.history".

### 3.2.2 Example 2: ECG device xAP enabled

Where a device like an ECG apparatus is used to measure a patient's condition, the full patient's database information is not required by the device. The device would simply maintain a list of enrolled patients ID numbers. The operator selects the patient from the list and performs the ECG as normal. The device would then on completion of the test broadcast the abbreviated results of the tests in the following format:

```
xap-header
{
v=13
hop=1
uid=TM.0002:0001
class=xAPBSC.event
source=clinic0002.ECG0001.7502185028080.Mosadi
}
ECG.info
{
MachineName=ECGmachine
Manufacturer=Siemens
Model=ECG05
LastCalibrated=20100101
}
ECG.test
{
TestType=Standard Cardiogram
TestDuration(min)=5
TestOperator=Mosadi
TestDate=20100301
TestTimeStart=1543
PatientAge=45
PatientBMI=5
SamplePeriod(millesec)=25
}
ECG.result
{
AverageHeartRate=85
MinimumHeartrate=67
MaximumHeartrate=165
OperatorsComment=patient was anxious and coughed often during the test
Graph=_____
TestURL1=ftp://ftp.tm.org.za/7502185028080/ECG/20100301/ECG1.jpg
TestURL2=ftp://http.tm.org.za/7502185028080/ECG/20100301.htm
}
```

The UID identifier "0001" refers to the device namely the ECG apparatus. The class of message is an event. The source is again clinic 2 where the tests were done, and the device followed by the patients IDnumber, and the person who operated it i.e. nurse Mosadi. The message area now has three message blocks ECG.info, ECG.test and



ECG.result, respectively containing the machine information and calibration, then the test type and parameters, and lastly the test results. The test results also contain a graph which could be created by the source device software in a byte or word format. A graph sent by text would be based on the 255 ASCII characters set sequentially in relation to the graph shape. The information that can be transferred is very basic, but could be deemed life saving in certain cases. Graph to text encoder and decoder subroutines would be required.

A URL for further information can be included which on the recipient side would appear as hyperlink to an ftp or http server. Details of the device test can then be retrieved when online. The location of this server, whether centrally or at the clinic would be based on the internet stability at the clinic. Often centralized server location and management is considered better, but when a clinic is offline, regular patients cannot be treated due to information held offsite. Servers based at the clinics and remote managed is recommended. The retrieval rate is increased and no offline conditions would cause the non treatment of patients. However, some patients move between clinics, and offsite data backup is considered important in all organizations.

Here xAP automation's natural mechanism of broadcasting any changes as they occur is helpful, as the central robots would seek new data only from the list of xAP sourced systems. When a clinic comes online, the central robots would receive a burst of xAP messages that it would then use to update the central systems. Where the xAP message includes a URL, the central robot would automatically retrieve the file, and store it on one or more mirror sites.

#### 4 CONCLUSION

The transfer of eHealth data via an intermitted datalink requires a robust protocol that can be buffered and retransmitted on request when the datalink has been restored. The proposed framework suggests a medical centre (source) broadcasting small messages to announce new or updated data on a patient. The small xAP datagrams can be transmitted over GPRS 3G or even SMS. Another medical centre (receiving) would update its database with merely the fact that information is available on that patient. If the patient checks into the receiving medical centre, it would have all information on the patient. However, it will not have all the bulk information. It would then retrieve the bulk information as required from the source medical centre.

The requesting medical centre would use a smart client to negotiate a data transfer format suitable to its device capabilities. Existing consumer level devices like smart phones and tablets are often adequate as human interfaces. Research on the transport of xAP messages to smart phones should be continued. Friendly smart applications have to be developed to provide efficient interface to the medical practitioner and patient alike.

#### 5 REFERENCES

- [1] Ferrer-Roca, O., Cardenas, A., Diaz-Cardama, A. & Pulido, P. 2004. Mobile phone text messaging in the management of diabetes, *Journal of telemedicine and telecare*, 10(5), pp 282.
- [2] Taluker A K, Das D. 2010, Mobile web for under-privileged in developing countries
- [3] *Telematics and Informatics* 27 (2010) 350-359
- [4] Van der Merwe A.F. 2010. xAP open source protocol fo telemedicine, *Proceedings of SAIE conference 2010*
- [5] Varshney, U. 2006. Using wireless technologies in healthcare, *International Journal of Mobile Communications*, 4(3), pp 354-368.