Taking Collaboration to the Next Level

“We have seen the future and it is here…”

Presented by
Alex Ho
Founder, CEO

WWW.ANABAS.COM   Tel: (1) 415.651.8808

by

Alex Ho, Anabas
Geoffrey Fox, Anabas/Indiana University
Motivation

• The number of mobiles exceeds the number of PCs
  • Samsung projects mobile market size of 1.2 billion units in 2008
  • Gartner reported about 271 million units of PC sold in 2007
• Information is playing an increasingly important role in mobile applications
• Increased use of low cost sensors in commercial and consumer environments
• Information age versus Integration age
  • Too much information. Need integration and relevancy.
• Increase interests in real-time collaboration for social networks
Global Mobile Survey Sponsored by Qualcomm & Telco 2.0

- Which companies have the most impact in shaping the future of mobile communications?

![Bar Chart]

- Google: 21.10
- Nokia: 16.10
- Apple: 13.45
- Microsoft: 8.86
- Vodafone: 4.49
- Ericsson: 3.77
- Qualcomm: 1.32
- China Mobile: 1.32
- Samsung: 1.27
- Huawei: 1.12
- Yahoo: 1.12
Quest for Dominance in Mobile Application Development Platforms

• Nokia Symbian – open
• Google Android - open
• Apple iPhone – restricted open

Open systems ensure
• application openness is maintained on the mobile Internet
• takes about 20% off a mobile’s cost (software)
• facilitates growth of third party applications to drive adoption
• This talk does not focus on what is and may be a killer app.
• We focus on providing a framework that facilitates a world of talented developers to build and deploy creative and potentially killer-applications easily on
  - low power, battery-based devices such as a mobile handset, and
  - an open platform that is designed for smart mobile devices such as Symbian
Understanding Application Framework Requirements from an Application Perspective
Global Mobile Survey Sponsored by Qualcomm & Telco 2.0

- What are the opportunities for growth in the mobile service and application areas?
Global Mobile Survey Sponsored by Qualcomm & Telco 2.0

• What are the opportunities for growth in the mobile service and application areas?
Some Sample Sensor-Centric Android Applications

• Accelerometer and Compass
  • Open Google Map’s StreetView functionality streaming over a 3G network
  • Using the built-in 3D accelerometer and compass, a user can take a “look” around a street and location simply by holding the handset and moving it around.

• Camera, GPS, Motion Sensor and web services
  • Lay and display location services over live imagery.
  • By pointing a camera at a building a user will be presented with the building’s name/location, and distance from the user.

• Camera and GPS
  • Social network around geographical content
  • Search for places to go, exciting routes
  • Create navigation by picture and share
Samples Collaborative Android Applications

• **Collaborative Painting**
  - Share a canvas and invite friends to draw
  - Draw on photographs

• **Instant Messaging App**
  - Location-aware mobility communication among friends
  - Interactive and synchronous map sharing

• **Jigsaw: Image-processed Whiteboard Sharing**
  - Utilizing image processing techniques like edge detection, geometric transformation and image enhancement to capture whiteboard data
  - Share the whiteboard and its data
Observations on mobile applications

• Interested in collaboration and sharing
• Sensor-Rich
• Access outside services like maps, music, TV, GIS
What do the observations really mean for mobile apps?

• **Social network is important.** A framework that has built-in support for comprehensive, effective and efficient collaboration will enable more addictive social networks.

• **Environmental sensor information from mobiles are of interest.** Mobiles are naturally globally distributed. The capability to harness and manage distributed sensor streams and at the same time supporting real-time sharing in an application development and deployment framework will substantially simplify and make it much easier for developers to focus on building the next Myspace-like killer-app by leaving the complex distributed deployment and information management task to the framework.

• **Take full advantage of the mobile network.** Not everything need to be done on mobiles for mobile apps. Some tasks are preferred and better done outside mobiles. Supporting easy access to outside services like maps, music etc is an important aspect of the mobile Internet. It will even be more compelling if mobile apps running on an Nokia-led mobile Internet could be enriched with transparent access to and interaction with any outside services, including the zillions of existing software capabilities that normally run on multi-cores or servers, in addition to supporting retrieval of multimedia data like other competitors.
Test Case 1: How about an interesting yet challenging cross-device mobile application? Can it be done easily?

- Do all the great things with mobile camera, GPS, maps, GIS, compass ….
- Capture the lovely view with other environmental data of the surroundings
- Add annotations and overlays
- Share it synchronously with not only social networks on mobiles but also devices like a digital picture frame in grand-parents house

Test Case 2: For the sake of stimulating thoughts and deriving architectural requirements, how would one develop an Android Jigsaw-like whiteboard application easily on Symbian?

Question: What is a viable, consistent architecture to support easy development and deployment of these test case applications and all others we looked at earlier?
“Many of us envisioned an online world where constellations of PCs, servers, smart devices, and Internet-based services can collaborate seamlessly. Business will be able to share data, integrate their processes, and join forces to offer customized, comprehensive solutions to their customers. And the information you or your business need will be available wherever you are - whatever your computing devices you are using.

That vision has not yet been achieved.”

- Bill Gates, 2005
Anabas Cross-device Collaboration Platform

Providing a “pluggable” architecture to enable rapid collaborative application development and deployment

ENABLING A NEW CLASS OF CROSS-DEVICE, INTERACTIVE, RICH MEDIA APPLICATIONS

Examples of possible vertical collaborative applications

- eLearning Virtual Classroom
- Anabas ClassTime
- Web Conferencing
- Anabas Impromptu
- Collaboration Application
- Collaborative Game Production
- Collaboration Application
- Collaborative Diagnostics
- Collaboration Application
- Multi-Participant Interactive Exercise
- Collaboration Application
- Collaborative B2B
- Other eCollaboration & eLearning Vertical Applications

Anytime, Anywhere Cross-Appliance architecture

Scalable & Reliable 24 x 7 ASP Distributed Server Architecture

Collaborative Application Management Framework
- Security
- Access Control
- System Management

Anabas Cross-device Collaboration Platform

ENABLING A NEW CLASS OF CROSS-DEVICE, INTERACTIVE, RICH MEDIA APPLICATIONS

Examples of possible vertical collaborative applications

- eLearning Virtual Classroom
- Anabas ClassTime
- Web Conferencing
- Anabas Impromptu
- Collaboration Application
- Collaborative Game Production
- Collaboration Application
- Collaborative Diagnostics
- Collaboration Application
- Multi-Participant Interactive Exercise
- Collaboration Application
- Collaborative B2B
- Other eCollaboration & eLearning Vertical Applications

Anytime, Anywhere Cross-Appliance architecture

Scalable & Reliable 24 x 7 ASP Distributed Server Architecture

Collaborative Application Management Framework
- Security
- Access Control
- System Management

Anabas Cross-device Collaboration Platform

Providing a “pluggable” architecture to enable rapid collaborative application development and deployment
Some Earlier Work in Cross-Device Collaboration
Some screen shots of Anabas/IU CGL collaboration demos.

1. On PCs, Sprint Treo 600, Compaq iPaq, Nokia 3650 and Polycom.
2. On PCs, Sprint Treo 600, Compaq iPaq, Sprint Samsung Sph i300.
3. On PCs, Nokia N800, Lego Mindstorm NXT robots and Webcam.
4. On PCs, RFID, GPS, Ultrasonic, Sound, Light and Touch and Webcam video sensors
Cross-device collaboration – Anabas/IU

(1) Figure a: An Impromptu collaboration client runs on a PC and shares with a Sprint Treo 600 handset and a Compaq iPaq PDA.

(2) Figure b and c: 3 Webcam streams and an animation stream being shared between a Nokia 3650 and Polycom device.
An Anabas Impromptu collaboration client runs on a PC and shares a presentation with a Sprint Treo 600 handset, a Sprint Samsung Sph i300 handset and a Compaq iPaq PDA.
Back to the questions:

How about a cross-device mobile application?

• Do all the great things with mobile camera, GPS, maps, GIS, compass ….
• Capture the lovely view of the surroundings with environment data
• Add annotations and overlays
• Share it synchronously with not only social networks on mobiles but also devices like a digital picture frame in grand-parents house

What is a viable, consistent architecture to support this application and all others?
Objectives for a Collaborative Sensor-Rich Application Development Framework

To enable easy

• development
• deployment
• management
• real-time visualization
• organization
• presentation

of collaborative sensor-rich applications.
Our definitions:

• A sensor is a time-dependent stream of information. A sensor is a resource. The sensor could be external, such as a TV stream.

• Grid computing enables any level of virtualization of resources so that users/applications do not need to know where the resources are before accessing to or interacting with them.

• A sensor grid virtualizes distributed, heterogeneous sensors allowing transparent access to, sharing of and interaction with any deployed sensors, any time, any where.
Introduction – Grid & Services

- Grids and Cyber-infrastructure have emerged as key technologies to support distributed activities that span scientific data gathering networks with commercial RFID or (GPS enabled) cell phone nets.
- Phoenix extends the Grid implementation of SaaS (Software as a Service) to SensaaS (Sensor as a service) with a scalable architecture consistent with commercial protocol standards and capabilities.
- Phoenix further extends the support of Software as a Sensor, in which case SensaaS includes Software as a Service in a unified framework.
Background

Commercial Backdrop

- XaaS or X as a Service is dominant trend
- X = S: Software (applications) as a Service
- X = I: Infrastructure (data centers) as a Service
- X = P: Platform (distributed O/S) as a Service
- Grids are any collection of Services and manage distributed services or distributed collections of Services i.e. Grids to give Grids of Grids
- We added
  - X = C: Collections (Grids) as a Service and
- X = Sensors as a Service
High-level Phoenix System Architecture

**Sensor Layer**
- Images
- RSS
- RFID Tags
- GPS

**Meta Data Layer**
- Location
- Logical Groups
- Sensor Type
- User Defined Properties

**Information Management Layer**
- Messaging Services
- Sensor Management

**Application Layer**
- Consumer Applications
- Logistics Applications
- Financial Applications

**Military Applications**
- Logistics Applications
- Financial Applications

**Logistics Applications**
- Military Applications

**Financial Applications**
- Military Applications

**High-level Phoenix System Architecture**

**ANABAS**
Sensor Layer
• Sensors provide raw information which is captured dynamically in different environments.

Metadata Layer
• Describes the properties of sensor; gives meaning to raw data collected from sensors. Makes information filtering possible.

Information Management Layer
• Transport messages from sensors to applications
• Messaging facilities that supports multi-protocol
• Facilities for sensor management such as deploying and disconnecting sensors

Application Layer
• Applications access to virtualized, distributed resources
Phoenix: A Collaborative Sensor-Centric Application Development Framework

Phoenix API allows application developers to retrieve sensor data and metadata about sensors. The Phoenix SSAL facilitates sensor developers to define sensor metadata for application-level filtering and exposes sensor services to applications.
Typical Sensor Grid Interface

- Different UDOPs
- Sensors Available
- Participants
- Presentation Area
A Sneak Preview of A Sample Collaborative Sensor-Centric Grid Demo Scenario

- The robots in the demo are Lego NXT Mindstorm robots.
- Each NXT robot carries some sensors – just like some new mobiles do.
Supported Services

Sensor Services:
- RFID
- GPS
- Wii remote
- Webcam video
- Lego Mindstorm NXT
  - Ultrasonic
  - Sound
  - Light
  - Touch
  - Gyroscope
  - Compass
  - Accelerometer
  - Thermistor
- Nokia N800 Internet Tablet

Computational Service
- VED (Video Edge Detection)
Distributed Architecture for Data Access
Illustrative Distributed Architecture for Data Access
(including a Nokia N800)
Data Model

• Sensors in different geo-spatial locations continuously publish data into the distributed brokering network.

• Phoenix routes relevant data to all connected applications according to their client-side requirements.

• Applications are notified for each data arrival.

• Some sensors are capable of receiving requests from applications and perform some actions in return.

• Sensors data could be routed to other computational services for further processing.
Data Selection and Filtering

• Each Phoenix client application is only interested in certain domain-specific information extracted from the large raw data pool supported by Phoenix.

• Filtering mechanism allows
  • an application user to define “filters”
  • a filter is sent to Phoenix
  • Phoenix responds with sensors data that match the filter
  • the application subscribes to data of these sensors through the Phoenix API
An example of a filter in Phoenix

A decision-maker wants to locate all GPS and RFID sensors in US or UK, the corresponding query looks like:

\[
\text{sensorType}=\text{GPS} \land \text{sensorType}=\text{RFID} \land \text{location}=\text{US} \lor \\
\text{sensorType}=\text{GPS} \land \text{sensorType}=\text{RFID} \land \text{location}=\text{UK}
\]
A sample Phoenix-capable sensor-centric application provides a GUI to support ease of filtering.
Grid Builder (GB)

GB is a sensor management module which provides services for:

- Defining the properties of sensors
- Deploying sensors according to defined properties
- Monitoring deployment status of sensors
- Remote Management - Allow management irrespective of the location of the sensors
- Distributed Management – Allow management irrespective of the location of the manager / user
Sensor Grid (SG)

Sensor Grid

Grid Builder

SSAL
  Common interface
  Device-specific communicator
  Interface (USB/Serial/Bluetooth)

Sensor Client program

Physical Sensor
A Collaborative Sensor-Grid Demo

-- An illustrative demo in CTS 2008
Supported Services

Sensor Services:

- RFID
- GPS
- Wii remote
- Webcam video
- Lego Mindstorm NXT
  - Ultrasonic
  - Sound
  - Light
  - Touch
  - Gyroscope
  - Compass
  - Accelerometer
  - Thermistor
- NOKIA N800 Internet Tablet

Computational Service

- VED (Video Edge Detection)
Recap

Test Case 1: How about an interesting yet challenging cross-device mobile application? Can it be done easily?

- Do all the great things with mobile camera, GPS, maps, GIS, compass ….
- Capture the lovely view with other environmental data of the surroundings
- Add annotations and overlays
- Share it synchronously with not only social networks on mobiles but also devices like a digital picture frame in grand-parents house

Test Case 2: For the sake of stimulating thoughts and deriving architectural requirements, how would one develop an Android Jigsaw-like whiteboard application easily on Symbian?

Question: What is a viable, consistent architecture to support easy development and deployment of these test case applications and all others we looked at earlier?
• High level approach to Test Case 1: Cross-device Sharing
  • Develop mobile phone as a SensaaS (just like an NXT robot).
  • Develop digital picture frame as a SensaaS.
  • Deployed both SensaaS on the Phoenix Sensor Grid.

• High level approach to Test Case 2: Jigsaw-like Whiteboard on Symbian
  • Use the Test Case 1 “mobile phone as a SensaaS”.
  • Add annotation and drawing features as Jigsaw.
  • Develop all image processing capabilities as SensaaS on Windows or Linux. Image processing algorithms could be developed much easier and run much faster on Windows or Linux PCs than Symbian on mobiles. These algorithms are most likely even exist in some source form. Just wrapped them as SensaaS and saving all the time to re-invent an optimized implementation on Symbian.
  • Leverage the Nokia mobile Internet.
  • Deploy all relevant SensaaS on the Phoenix Sensor Grid.
  • SensaaS streams can be shared synchronously without additional coding.

Conclusion
Conclusion

• Phoenix provides a unified architecture to support Collaborative Sensor-Rich applications by treating “XaaS” uniformly. It supplements Connecting People with Connecting Resources.

• Phoenix enables a new class of sensor-rich, streaming mobile application development, deployment, management and delivery platform for leading mobile service providers. It enables mobile service providers to serve all the mobile streaming needs of their end users and application developers.

• As mobile sensor streaming traffic traverses the Phoenix sensor grid, a StreamingTube™ SensaaS could be added to support storage and asynchronous access to the world of streaming content that originates from or is destined for mobiles.