Iowa State University

Lost in the Information World
The Janix System

CSIDC 2005: Final Report

Team Members:

Andrew Lundberg
Christopher Hagen
Janice Kar Bo Wong
Chun Yu

Project Mentor:

Arun Somani

1) Abstract

In a world that is becoming increasingly mobile, people are demanding real-time information distribution. However, extracting and making available only the most useful and relevant information on a device that is available to almost all users, such as a Bluetooth capable cell phone, can be challenging. The goal of the Janix (Janus Adaptable Network of Integrated Cubix) system is to define, design, implement and demonstrate a new wireless communication system to retrieve such information. The system uses a two-layered wireless network that is scalable to several different applications and environments. For example, it could be deployed to provide sales information in a shopping mall or convenient lookup of flight times, luggage status, media information, or sports scores in an airport. By delivering relevant information to each user on demand, they will no longer be lost in the information world.

Possible deployment scenario: Consider the following scenario in which the Janix system improves the overall experience of its users.

After being fed up with defamatory articles written by the noted travel journalist Reggie Black, the owners of JARC International Airport decided to improve the experience of their customers in an effort to increase business. They decided to implement the Janix system because of the functionality it would provide to both travelers and employees. The Janix software developers worked closely with airport employees to develop applications that could track luggage, give directional assistance, provide entertainment information and distribute flight schedules. Once the system was in place, the overall atmosphere of the airport was improved.

Upon entering the terminal at JARC international, Reggie was expecting another bad experience that would provide plenty of juicy information for a new article. Seconds later he was surprised as his Bluetooth capable cell phone connected to the airport's Janix system. He was immediately welcomed to the system and was surprised by the different information provided to him. Reggie immediately used the system to check the status of his flight; seeing it was delayed, he decided to spend some time in the lounge exploring the new service provided by the airport. Reggie's amazement grew as he accessed special sales information, checked the status of his luggage and found that his favorite team had won the WEPA championships. After returning home, Reggie wrote a positive article preaching the abilities of the new system implemented by JARC informational airport.

Socially Useful Functionality: The true strength of the Janix system lies in its flexibility and lack of required user registration. It provides a wireless system that can be configured to meet the requirements of many situations, going beyond the boundaries of any system, for example:

- *Airport* – The Janix system could be used to provide flight schedules, luggage status and shopping information.
- *Medical* – The Janix system could be used to monitor patient's vital signs as they are moved throughout a hospital, allowing doctors to monitor progress any change in their patient's status.
- *Libraries* – The Janix system could be used to direct patrons to books, and locate misplaced books using embedded sensors similar to those used to prevent theft.

Janix opens up several avenues in the area of distributed sensor processing applications.

2) System Overview

The Janix system is made up of three separate subsystems as seen in the following diagram: Janix node, host computer and Bluetooth capable user devices.

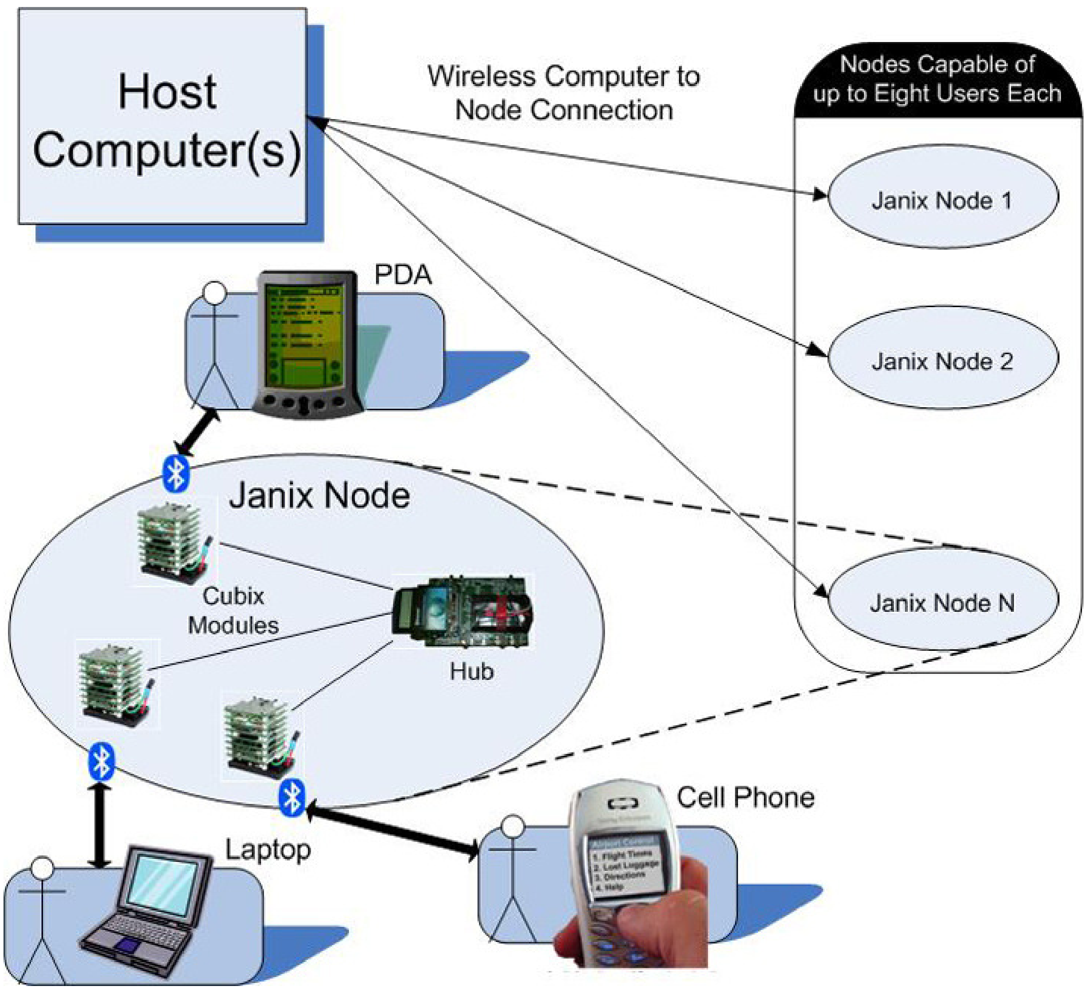

Figure 1: Schematic of the Janix System

In order for the system to provide reliable connection to all users, a communication algorithm that used the included technology to its fullest potential was required [10]. A typical situation, such as the possible airport deployment scenario outlined above, would work as follows:

1) During the initial phase the host computer connects to the hub through WiFi and prepares itself to receive a list of devices.
2) The hub discovers all the Bluetooth capable devices inside a ten meter range, these include both the Cubix modules and any Bluetooth capable user devices that come in range during this discovery phase.

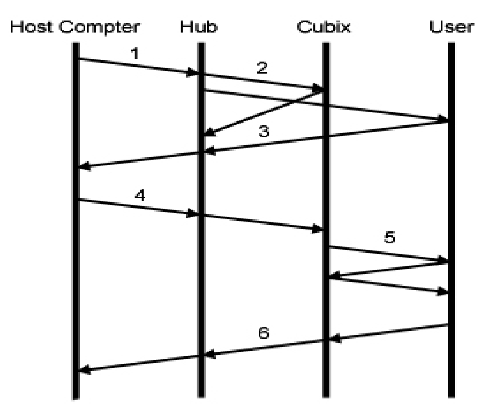

Figure 2: Communication Diagram

3) The hub provides the host computer with a list of Bluetooth capable user devices and Cubix modules that are inside its range. The host machine parses this list in order to find all of the user devices.
4) The host computer then sends a command to an idle Cubix via the hub.
5) The Cubix then connects to the user's device and sends the proper data, i.e. user menu. The user interacts with the menu system provided by the Cubix. This allows them to receive programmable menu options and real-time sensor data.
6) The user exits the system causing the host computer to release the user from the list of connected devices, allowing the user to reconnect to the system.

Once the first four steps have been completed, the host computer returns to the first step and reruns the process. If a user is still connected to a Cubix during the Bluetooth discovery phase, the hub will be unable to discover it. This keeps the host computer from attempting to use any device that is in use.

Design Methodology

A top down design approach was followed while developing the Janix system [8]. This method was chosen after examining the capabilities of the hardware being used. The high level concept for this project was to develop an information distribution system. This system would need to be able to communicate with several different types of wireless devices, provide user interaction and be capable of monitoring its status.

In order to accomplish this goal, small scale sample applications were developed to test the capabilities of the hardware and the software development language. While these programs were being written, the overall distributed network concept was envisioned. Once the network architecture had been developed, the team began developing three separate software suites: Cubix module suite, host computer control suite and Bluetooth capable applications suite.

Innovative Concepts

The Janix system is a unique combination of Bluetooth and WiFi technology, which can be used to navigate a user through the information world. It overcomes the following weaknesses inherent in other wireless information distribution systems.

- In many systems utilizing cell phones, users are bombarded with spam.
- Many systems require users have vendor specific hardware and software, such as specific wireless service plans.
- A high resource overhead is required to change any functionality in the system.
- Often times, the provided information is outdated.

The Janix system blends the strengths all the included technologies in order to overcome the above weak points.

- Janix can be upgraded easily, allowing implementers to add new functionally.
- The Janix user centric approach allows users to specify the content they receive.
- The list of supported devices of a Janix system can easily be increased.
- The system's flexible architecture allows for easy implementation.
- All information in the Janix system is gathered and sent in real-time.

3) Implementation and Engineering Considerations

This section provides an in-depth description of current technology being used to create the Janix system.

3.1) Hardware Engineering

This section covers the hardware that makes up the Janix system.

Janix Node

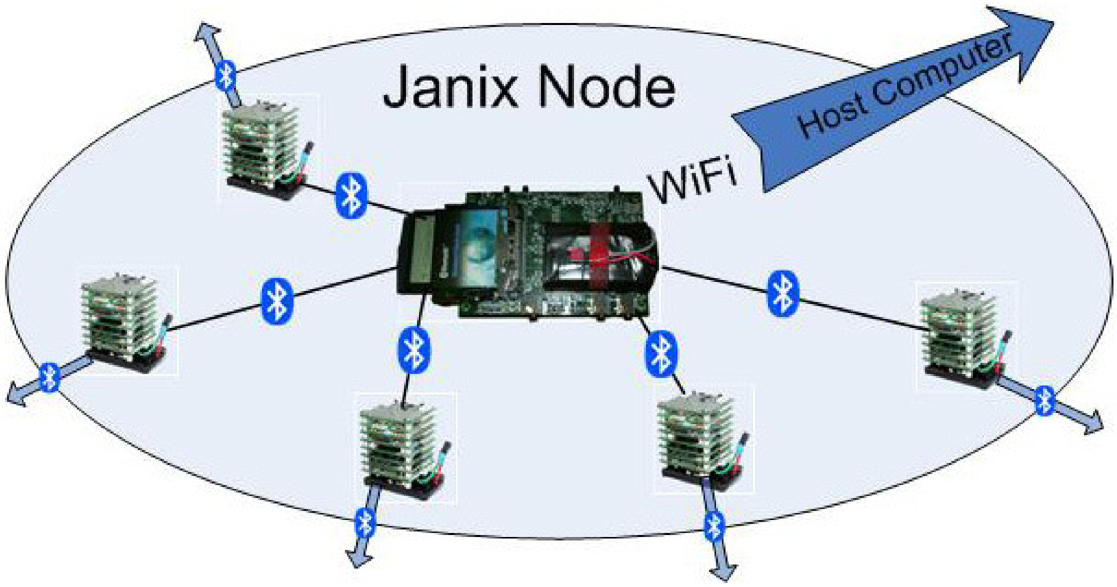

Figure 3: Schematic of a Janix Node

The Janix node is the main component of the Janix system. It is comprised of one two-layered, Bluetooth to WiFi, hub and up to eight Cubix.

- *Two-layered hub* – The hub provides a communication bridge between the IEEE 802.11b wireless standard used by the host computer and Bluetooth used by the Cubix modules [3].

Figure 4: Two-layered Hub

- *(Codename) Cubix module* - A small, battery powered, embedded computer system currently being developed by National Instruments. It provides wireless networking and data sensing capabilities, through added sensors. Cubix is designed to be a modular system made up of eight different boards that can be changed depending on the needs of the application. The current research model includes the following component as seen in figure 5 at the top of the next page [4].

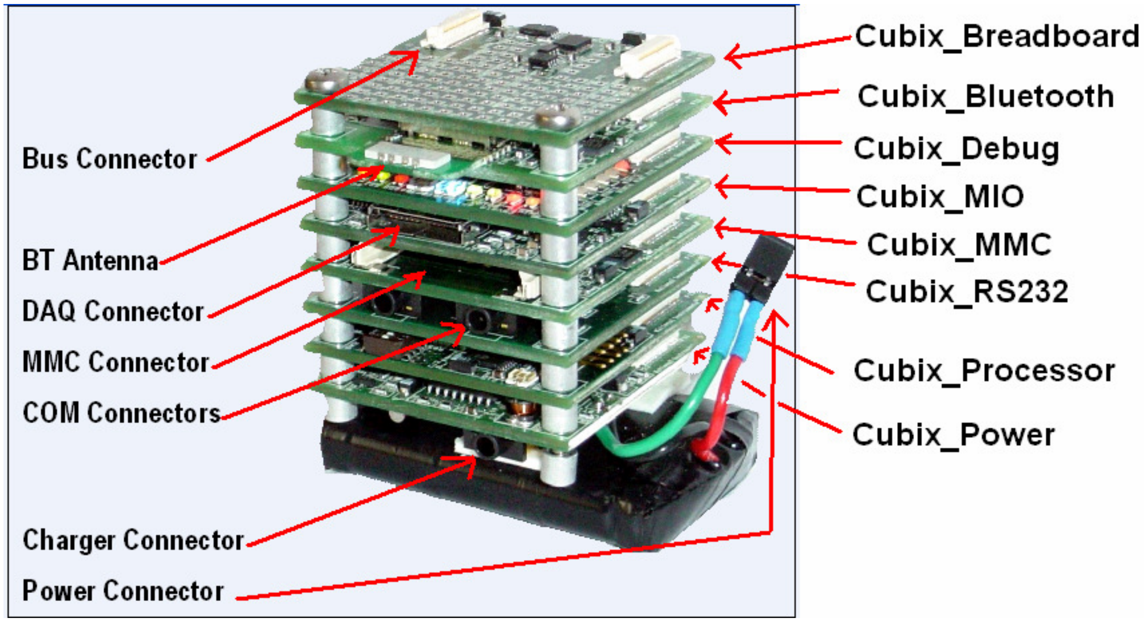

Figure 5: Cubix Module

Current List of Sensors Supported by the Janix System - The following is a list of sensors that have been integrated into the Janix system:

- Temperature and humidity - Sensirion SHT11 [9]
- Passive infrared - Passive Infrared Movement Detector Kit (CK206) [1]
- Light detection - CdS Photoresistor Multi-Pack [7]
- GPS - WAAS-enabled Deluo GPS for Laptops and PDA's [2]

Bluetooth Capable User Device

In the Janix System, information is distributed to users through Bluetooth capable devices. For example, users can look up flight information from their Bluetooth capable cell phones when they are connected to a Janix System installed in an airport.

When a Bluetooth device is detected by the system, the Cubix asks the user if he/she wishes to pair with the Janix system [4]. If the user accepts the request, a user menu is created and uploaded to the Cubix. Once the Cubix has received this menu it begins interacting with the user. However, if the user rejects the request, they will not receive any messages from the Janix system.

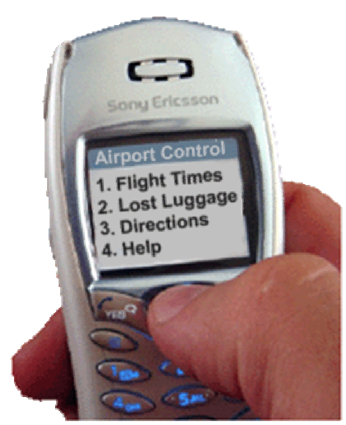

Figure 6: Sample Phone Menu

3.2) Software Engineering

This section provides more information about the software that was used in and developed while creating the Janix system.

Software Used

The following section describes the software used to develop the Janix system.

LabVIEW – All the software developed to run the Janix system was created using LabVIEW for Embedded Systems, a graphical programming language engineered by National Instruments [5]. LabVIEW uses the concepts of Virtual Instruments (VIs) as building blocks to create complex applications, as can be seen in the hierarchy diagram as show in figure 7, which represents a Bluetooth communication virtual instrument. Each block in the hierarchy diagram represents a separate virtual instrument that is loaded at run-time. VIs are created using a set of library functions that allow developers to build their own virtual instruments. For example, the picture below is a virtual instrument that is used to add each Cubix to the graphical user interface based on the sensors that are connected to it.

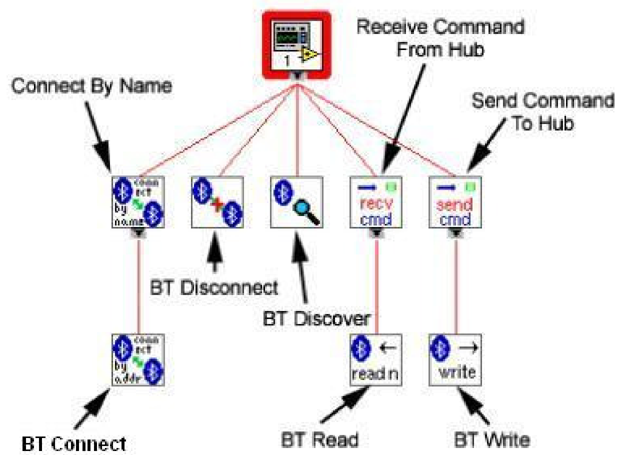

Figure 7: Hierarchy Diagram

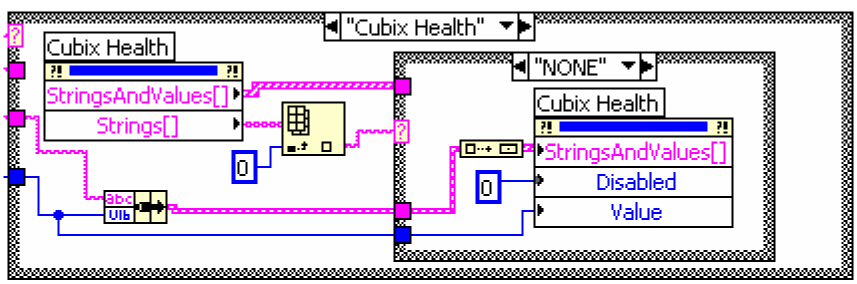

Figure 8: Sample Virtual Instrument

Rand McNally Street Finder – This software is used to display customized maps from GPS coordinates downloaded from a Cubix that is connected to a GPS receiver [3].

Software Developed

The following section outlines the issues surrounding the software developed for the Janix system.

Distributed Computing and Resource Management - In order for the Janix system to be implemented in a large scale application, such as an airport, the amount of data the host computer receives needed to be minimized to conserve bandwidth. This was achieved by distributing the system communication needs to the individual Cubix modules as much as possible. Once a user connection has been established, the Cubix module is solely responsible for the user interaction leaving the host computer free to do other things. This transfer of user interaction control can be seen in the following state diagram.

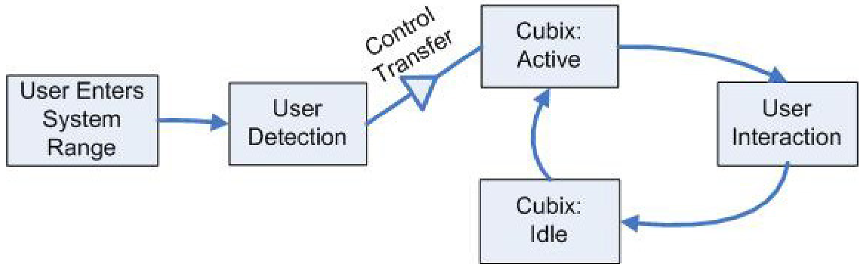

Figure 9: Connection State Diagram

The algorithm implemented to achieve this distribution relies on the principle that a Cubix is invisible to the host computer while a user is connected to it [4]. More details about the protocol developed for the distribution algorithm was outlined in the system overview.

Since the Cubix modules may be required to operate on battery power for long periods of time, an algorithm was developed to minimize the amount of computing required on each Cubix [4]. Given that a cell phone and a Cubix have very little computing power when compared to a PDA/laptop, the optimization was implemented in the software to be run on the PDA and laptop. This optimization was accomplished by passing the computing power needed to interpret the Janix menu system to the software on these two devices. Further battery power is conserved by having the Cubix go to an idle state when no users are connected (as seen in figure 9).

Sample Application Sequence - The following diagram represents a possible sequence of events that could occur in a Janix enabled airport.

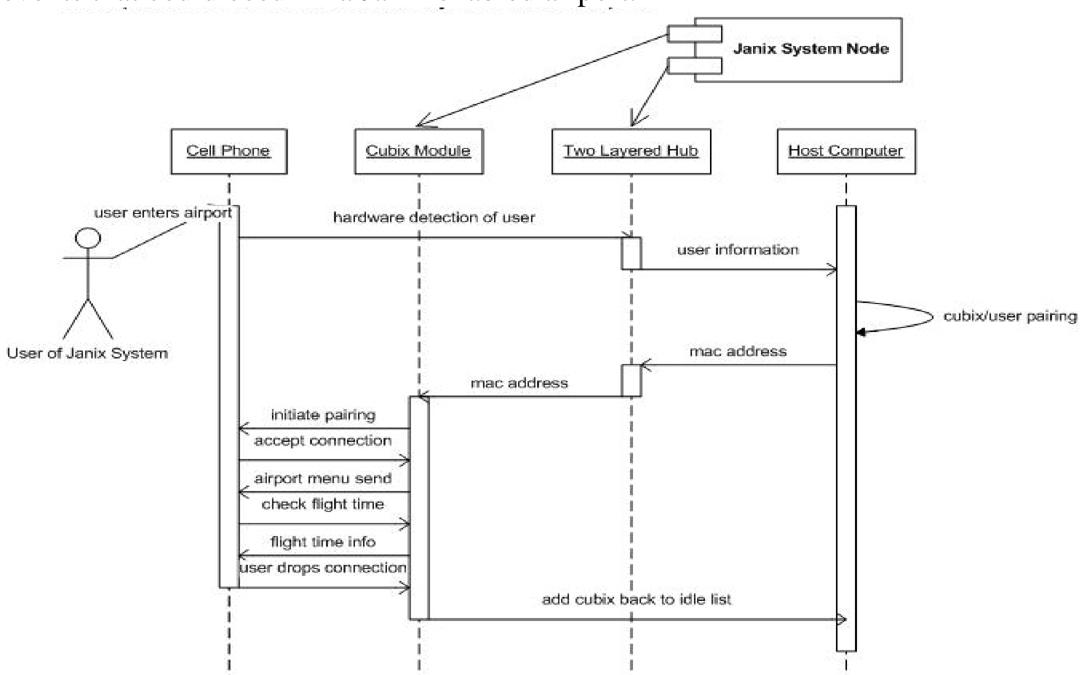

Figure 10: Janix Sequence Diagram

In order to provide the functionality shown in figure 10, the team designed each of the following three software suites to include virtual instruments that could provide the necessary capabilities.

Cubix module suite – This includes all virtual instruments that were developed to run on the Cubix, except for those which interact with Bluetooth capable devices. These virtual instruments include the following:

- *Cubix sensors* – These virtual instruments are developed to be capable of reading information from all of the sensors connected to each individual Cubix.
- *Data storage* – Certain applications, such as GPS tracking, may require that the Cubix store data. The data storage virtual instruments provide read/write functionality for the Cubix.
- *Host computer communication* – The Janix system implements a customized application layer protocol (ALP), as seen in table 1, in order to control the entire system. The Cubix are pre-programmed to receive the following commands from host computer and initiate the requested action. However, new commands can easily be added to the Janix system in order to increase functionality.

Table 1: Application Layer Protocol

Command	Input	Action
mac	Users MAC address and privilege level	Cubix connects to the device specified by the MAC address
menu	Menu data	Inputs a menu hierarchy onto the Cubix
batt	NONE	Sends battery voltage from the Cubix
home	NONE	Sends list of Cubix capabilities to the control computer
sensor	NONE	Queries Cubix containing sensors inside the Janix node
motion_on	NONE	Turns on motion sensor connected to the Cubix
motion_off	NONE	Turns off motion sensor connected to the Cubix
gps_on	NONE	Start collecting data from the GPS receiver
gps_off	NONE	Stop collecting data from the GPS receiver
gps_interval	Number of minutes	Set the interval for polling coordinates from the GPS receiver

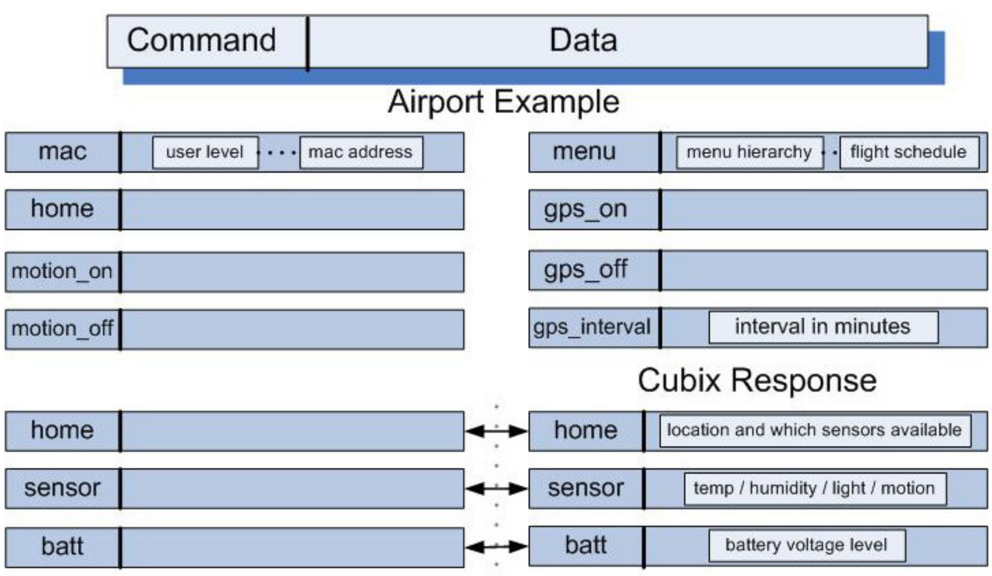

Figure 11: Example Data Packets

- *Build menu* – In order to interact with Bluetooth capable devices, each Cubix module in a Janix node must have a menu hierarchy loaded into memory every time a new user is connected. The menu command in the ALP initiates this VI and stores the data sent from the host computer into memory.

Host Computer Control Suite – This includes all virtual instruments that run on the host computer

- *Bluetooth discovery* – The host computer must be able to detect the presence of both the Cubix nodes and the Bluetooth capable user devices. This VI repeatedly polls the area, searching for new user devices and usable Cubix modules.
- *Device address parsing* – The host computer uses this virtual instrument to sort the Cubix and user devices into two separate lists.
- *Application layer protocol control (ALPC)* – The host computer provides the system administrator with the ability to send commands defined in the application layer protocol. This VI also automates the connection control process between a Bluetooth capable user device and a Cubix. Once the user's device has been added to the list of Bluetooth capable devices, it is connected to the first available Cubix.
- *User privileges* – This VI matches an inputted MAC address against a pre-determined list. If the address is found in the list, the corresponding privilege level is outputted. If it is not found, the default user level is outputted. The privilege level is added to the data packet that is sent by the ALPC the "mac" command.
- *Sensor* – The sensor virtual instrument polls the Cubix in the Janix system that contains sensors at one minute intervals. If that Cubix is currently in use, it waits for it to become idle.
- *Build user menu* – This virtual instrument reads a file that contains AT command set programming, shown in a table 2, and data [6]. The output from this file is used to build a string that represents the Bluetooth capable device menu. This allows the Janix system to build dynamic user menus.

Table 2: Sample AT Command Set

Command	Action
AT*EAM	Sends main menu to the Bluetooth capable user device
AT*EASM	Sends sub menu to the Bluetooth capable user device
AT*EAID	Sends a variety of dialog boxes to the Bluetooth capable user device
EAII	Reply from Bluetooth capable user device
EAAI	Sends selected menu option from the Bluetooth capable user device to the Cubix

- *Cleanup and error detection* – The host computer must be capable of removing old user devices and detecting errors in expected Cubix behavior, i.e. dead battery. This virtual instrument handles this task by comparing the output of the Bluetooth discovery VI with the current list of devices and computes the expected output. The system administrator is alerted to any errors with a pop-up window.

Bluetooth Capable Applications – This includes all virtual instruments that interact between the Cubix and the Bluetooth capable user devices.

- *Connect to user* **–** This virtual instrument receives a MAC address from the host computer and connects to the corresponding user device.
- *User interaction* – This virtual instrument uses the AT command set, shown in table 2, to communicate with the connected user device. It is capable of sending and receiving data, supplying the user the requested information.
- *AT command set interpreter* – Virtual Instrument that extracts pertinent information from an AT command set menu string [5].
- *User computer and personal digital assistant executable* – A VI running on either a Bluetooth capable PDA or computer, which receives input from the AT command set assembler, and provides user interaction with the Cubix. In order to accomplish this, the VI has an encapsulated AT command set assembler that has the ability to communicate with a Cubix module.

Computer and Personal Digital Assistant Menu Deployment

Bluetooth capable computers and PDAs must have the appropriate Janix menu executable installed in order to use the Janix system. There are several different methods for deploying these programs; choosing which to use is application specific. The programs could be distributed on removable media, such as compact discs, downloaded from a website or received from special onsite docking stations. In the airport example, it would be practical to provide users with the ability to download the programs from a website and receive from strategically placed docking stations, ensuring full coverage for software distribution.

3.3) Design Tradeoffs

Throughout the development process, many constraints and limitations were taken into consideration:

Cubix power consumption is an issue that has a major influence on the development of virtual instruments [4]. Since the Cubix may be deployed in an environment that has no AC outlet, the Cubix may have to run on batteries. Because of this, much of the heavy workload must be performed by the host computer and Bluetooth capable devices in order to preserve power and achieve maximum life span.

Due to the type of Bluetooth chip used on the Cubix, only one Bluetooth capable device may communicate with a Cubix at any time [11]. Such a limitation led to the decision of having multiple Cubix serving one area. Also, a timeout mechanism is used to disconnect idling users so that others may access the services provided by the Janix system.

The two-layered hubs used in the development of the Janix nodes can only connect to sixteen devices concurrently [4]. In order to meet the performance requirements of the Janix system, and stay within these limitations, it was decided by the team that the Janix nodes would only contain eight Cubix. Limiting the number of Cubix helps to lower network congestion, and host computer response time.

In exchange for an increase in size and complexity of the embedded programming, the flexibility and adaptation requirements of the Janix system were preserved. Since changing the system to deploy in different scenarios requires no changes to the programming, the administrator can focus on content creation.

The Janix system currently uses an extended AT command set that is supported only by Sony's Ericsson cell phones [6]. This command set allows menus and dialog boxes to be created via Bluetooth serial communication. In the future, support for other cell phones that use different command sets, with menu creation capabilities, may be added to reach a wider range of users. In such a situation the host computer will be responsible for examining the first three octets of the Bluetooth capable device in order to choose which menu to send.

3.4) Tools Developed for the Project

During the development process, several tools were created to facilitate the prototyping of concepts, testing of hardware, and debugging of software. To aid in prototyping of concepts, generic virtual instruments were created which could be reused and adapted to fit different applications. They contain commonly used code, such as communication port initialization, to allow rapid development for proof-of-concepts and feasibility studies.

For testing hardware components, a variety of simple applications were developed. The test programming was designed to treat the hardware being tested as a black box. A set of inputs is entered and the output is matched against the expected result.

Debugging software and hardware problems in a heterogeneous system can be a very frustrating and time consuming task. Special applications used primarily for debugging were developed to reduce the complexity and time consumption involved in the diagnosis of the system. The special Cubix debugging virtual instruments dump raw data to a reliable computer programmed with data capture VIs. The raw data dumps can be tailored to contain a variety of data, ranging from sensor data to wireless traffic. This allows the programmer to analyze incoming data quickly.

3.5) Verification and Testing

A modular approach was used in testing the Janix system [8]. Each program, sensor and Cubix node was tested separately. They were initially tested using electronic meters and test equipment. After confirming that all of the hardware worked properly, simple LabVIEW applications were developed to test each sensor on the Cubix. This ensured that each individual part of the system was functional before becoming integrated into the rest of the system. After verifying that all of the individual pieces of hardware were functioning properly, development of the Janix system began. Several tests were run after every milestone in order to ensure that each new part of the system worked before it become integrated, and that it did not adversely affect the rest of the system [8]. Using this approach lowered debugging time and helped to increase overall project efficiency.

Testing of a user oriented system is challenging, because finding code errors is not the only worry. It is also extremely important to verify that the software provides all functionalities required by the users. This is why it is important to have multiple methods of testing for each part to verify the system is both functional and usable. The following are examples of a few cases that were tested to insure that system functioned properly:

- Bluetooth capable device other than cell phone, PDA or laptop attempted to connect to the system
- User's dropping off unexpectedly
- Proper sample menu deployment
- Gathered sample sensor data

Full System Usability Testing - Once a Janix node was developed, a real world use case scenario, airport, was written in order to more thoroughly test the prototype [8]. It was designed to provide a simulated real life scenario in order to test the systems usability and functionality. The prototype Janix system was set up with five Cubix and a host computer. Five Bluetooth capable devices, two cell phones, one PDA and two laptops, were placed into the virtual environment.

With a simulated environment in place, the system was tested for software bugs and usability. After ensuring that the prototype test environment was stable, two different groups of third party testers were invited to help us test the system.

- *Test group type one* - These testers were people who are frequent flyers. An attempt was made to get both technically and non-technically oriented people.
- *Test group type two* – These testers were relatively unfamiliar with airports.

Groups of five were created from a mixture of both groups. They were provided with a Bluetooth capable device and asked to test the system. Individual feedback was registered from each test group, and applied to creating a better system.

Full System Maintainability Testing - After making several improvements based on the usability testing, it was decided that the airport use case scenario should be set up a second time in order to test issues dealing with system maintainability. Among these, the following were of primary concern.

- *Accountability* – In order to ensure maintainability, each of the three parts of the Janix system must be integrated in such a way that system administrators can be alerted to erroneous events.
- *Replacement* – Replacing bad hardware in the Janix system should not affect the operation of the system.
- *Intra-system independence* - It is extremely important for each piece of the system to continue functioning if some portion has a malfunction. This helps to provide accountability and maintains system availability

Once again the two group types of testers will be asked to test the system. While the system is being tested team members will purposely cause parts of the system to fail, these will include:

- Cubix modules will be turned off
- User devices will be turned off
- New menus will be created for deployment
- The host computer will be restarted unexpectedly
- The two-layered hub will be restarted unexpectedly

4) Practical and Deployment Considerations

This section covers issues involving deployment of the Janix system.

4.1) Janix wireless Security

Security is an important issue in any network. Due to the lack of security features in the Bluetooth protocol used by the Janix system, it is extremely difficult to ensure that data being transferred through the network is safe [4]. Janix is not designed to distribute sensitive data; anyone using it in this manner does so at their own risk. Even if a malicious user was able to gain access to a Cubix, he/she could not reprogram it; all they could do is receive the data that specific Cubix has or program a new menu into the Cubix. This would not cause any real damage to the Janix system because the menu is refreshed each time a Cubix services a new user. However, a user with a high level of knowledge about the Janix system could play a man in the middle attack between the host computer and a Janix node, or by hijacking a Cubix module from the system.

Regina White was a Janix administrator at JARC international, until she was fired for tardiness. She vowed to get back at her boss, formulating a plan that would discredit her former department. Regina went to the airport with her Bluetooth enabled Laptop and sat in the lounge discovering all of the Cubix in range. She opened the LabVIEW program that she had written the previous night which would mimic the host computer and two-layered hub, allowing her to send commands to an individual Cubix. After discovering an idle Cubix she immediately connected to it, and sent it the "menu" command which reprogrammed the Cubix with a new menu that encapsulated a bomb threat message. Regina then sent the Cubix the "mac" command, causing the Cubix to send the bomb threat to a user of her choice.

Fortunately, this type of attack can be monitored by the host computer, which will be able to discovery that the Cubix which Regina has hijacked is in use when it should be idle. This would cause an alarm to be sent to the system administrator. In order to increase the security of the Janix, certificates to authentic the host computer could be implemented in the future.

4.2) Marketability

The Cubix hardware and software being used by the Janix system is still under development by National Instruments. Neither an estimated cost, nor a possible release date has been announced. Due to these factors it is difficult to estimate any kind of marketability for the Janix system. Due to the limitation in range of Bluetooth technology, purchasing the Janix system is going to require a large initial investment [11]. This large cost can be lowered by using two-layered hubs that can connect to more than sixteen devices simultaneously and replacing the current Bluetooth chips used by the Cubix with those that can connect with multiple devices. This would increase the maximum number of users being serviced at one time, allowing implementers to use less than eight Cubix in each Janix node reducing the cost of the system. The design of the Janix system lends itself to be marketed as a product instead of a service. The ease of use and implementation would cause the manufacturer of the Janix system to be unable to charge a large monthly fee.

The table below shows the individual cost of each sensor currently implemented into the Janix system. Purchasing these devices for an actual implementation of the Janix system would be more cost effective, because their unit cost drops proportionately with volume ordered. However, many implementers are going to want to use more powerful, and more expensive, or even different types of sensor devices

Table 3: Cost of Currently integrated Sensors

Item name	Cost
Deluo GPS receiver	$74.90
SHT11 temperature and humidity sensor	$21.51
Passive Infrared Movement Detector Kit	$22.00
Cds Photo resister Multi-pack	$2.79

5) Summary

In today's high-speed data-dependent world, it is extremely important to have access to as much pertinent information as possible. The Janix system allows a system implementer to establish a wireless network that can streamline data distribution without requiring complex coding or expansive hardware knowledge. The system is extremely flexible; the same hardware and software components can be used for different applications with little adaptation. The Janix design helps to lower project costs and allows for a much larger user base.

Currently, the team has developed two Janix nodes using three Cubix for the first node, and two Cubix for the second node. The software suites have been prototyped to allow for testing several of the ideas that were introduced during the early stages of development; packet structure, host computer control abilities and system timing. The present version of the developed software supports all the functionality of the finished product, which provides users with the following functions:

- Connect to the system via Bluetooth capable devices
- Provide multiple levels of access based on user privileges
- Ability to view sensors connected to Cubix modules
- Ability to provide control to the system through the host computer and Bluetooth capable user devices
- Deploy different menus based on system administrator's choice
- Deploy new menus during runtime via the host computer

This functionality represents the backbone of the Janix system. However, because it is programmed using LabVIEW, new functions could be added at anytime, based either on use case or new technology. The following is a list of possible added features and hardware that have been discussed as possible additions:

- Build a WiFI board for the Cubix so that it can alert the host computer via ad-hoc network if the there is any errors with the Janix node two-layered hub
- Program the host computer to receive calls over a modem, in order to allow a system administrator to dial into the system from offsite
- Add to the list of sensors connected to the Cubix
- Expand the list of supported Bluetooth capable devices
- Replace the Bluetooth chip on the Cubix with one that allows multiple concurrent connections
- Add new security protocols to the system

The Janix system can provide its users with an extremely flexible information environment. It may be unclear how this product may be marketed, but the prototype has proven itself as an able guide in the information world.

References

[1] Carl's Electronics Inc. <http://www.electronickits.com/kit/complete/surv/ck206.htm>

[2] Deluo Electronics < http://www.deluoelectronics.com>

[3] Ghercioiu, Marius. "The Cubix, Getting Started Guide". June, 2004.

[4] Ghercioiu, Marius. "Cubix Manual". June, 2004

[5] National Instruments. "LabVIEW". 2004. http://www.ni.com/labview

[6]R320 AT Command Online Reference <http://linuxbrit.co.uk/downloads/R320AT_R1A.pdf>

[7]Radioshack. <http://www.radioshack.com/product.asp?catalog%5Fname=CTLG&product%5Fid=276-1657>

[8] Sommerville, Ian: Software Engineering 6th Edition. Addison-Wesley, 2000.

[9] Sensirion. <http://www.differential-pressuresensor.com/en/sensors/humidity/sensors_devices/sensorSHT11.htm>

[10] Stallings, William: Data and Computer Communications. New Jersey: Prentice Hall, 1996.

[11] The Official Bluetooth Website < http://www.bluetooth.com/>