ROVER TECHNOLOGY : ENABLING SCALABLE LOCATION-AWARE COMPUTING

Sahil Garg, Navdeep Kaur

[1]Student, Department of Mechanical Engineering, Chandigarh University, Gharuan, Punjab(India)

[2]UILA, Chandigarh University

gargsahil3600@gmail.com

ABSTRACT:

Rover stands for Remotely Operated Video Enhanced Receiver. Rover is next generation of Bluetooth, infrared & cellular services. Rover technology can be ranging from power full laptop to simple cellular phones. The technology which enables the scalable location aware computing. This involves automation availability of information & services based on current location of user. This user make avail location aware computing through his PDA.PDA is a hand held computer used as palmtop computer. PDA's commonly have colour screen & audio capabilities to be used as mobile phone, web browsers. Many PDA's can access the internet, intranet & extranet via Wi-Fi. Many PDA's employ touch screen technology.

Introduction:

Rover stands for Remotely Operated Video Enhanced Receiver. Rover is next generation of Bluetooth, infrared & cellular services. Rover technology can be ranging from power full laptop to simple cellular phones. The technology which enables the scalable location aware computing. This involves automation availability of information & services based on current location of user. This user make avail location aware computing through his PDA.PDA is a hand held computer used as palmtop computer. PDA's commonly have colour screen & audio capabilities to be used as mobile phone, web browsers. Many PDA's can access the internet, intranet & extranet via Wi-Fi. Many PDA's employ touch screen technology.

Theme:

We believe that Rover Technology will greatly enhance the user experience in a large number places, including visits to museums, amusement and theme parks, shopping malls, game fields, offices and business centers. The system has been designed specifically to scale to large user populations. Therefore, we expect the benefits of this system to be higher in such large user population environments.

We will use a museum tour application as an example to illustrate different aspects of Rover. We consider a group of users touring the museums in Washington D.C. At a Rover registration point in a museum, each user is issued a handheld device with audio and video capabilities, say an off-the-shelf PDA available in the market today. Alternatively, if a user possesses a personal device, he can register this device and thus gain access to Rover. The devices are trackable by the Rover system. So as a user moves through the museum, information on relevant artifacts on display are made available to the user's device in various convenient forms, for example, audio or video clips streamed to the device. Users can query the devices for building maps and optimal routes to objects of their interest. They can also reserve and purchase tickets for exhibitions and shows in the museum later in the day. The group leader can coordinate group activities by sending relevant group messages to the users. Once deployed, the system can be easily expanded to include many other different services to the users.

4.Rover services: (A) Basic data services. Rover enables a basic set of data services in different media formats, including text, graphics, audio, and video. Users can subscribe to specific data components dynamically through the device user interface. Depending on the capabilities of the user's device, only a select subset of media formats may be available to the user. This data service primarily involves one-way interaction; depending on user subscriptions, appropriate data is served by the Rover system to the client devices.

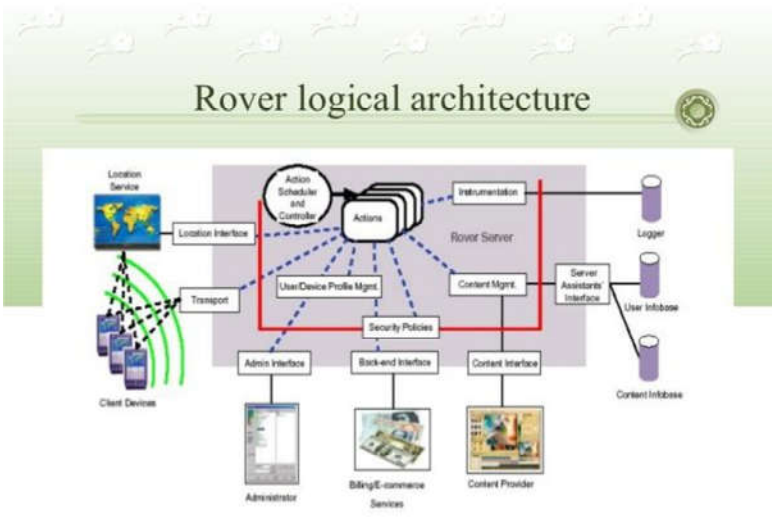
(B) Transactional services**:** These services have commit semantics that require coordination of state between the clients and the Rover servers. A typical example is e-commerce interactions. Services that require location manipulation are a particularly important class of data services in Rover. Location is an important attribute of all objects in Rover. The technique used to estimate the location of an object (some techniques are described in the Appendix) significantly affects the granularity and accuracy of the location information. Therefore an object's location is identified by a tuple of Value, Error, and Timestamp. The error identifies the uncertainty in the measurement (value). The timestamp identifies when the measurement was completed.

(C) Map-based services: Map-based services are an important component of location manipulation services. Rover maps can be subject to various operations before being displayed to users.

(D) Filter: Objects in a Rover map have a set of attributes that identify certain properties of the objects. Depending on the user's context (which indicates the user's interests), filters are generated for the attribute values of interest to the user. These filters are applied to maps to select the appropriate subset of objects to display to the user. For example, one user may be interested in only the restaurants in a specific area, while another user needs to view only the museum and exhibition locations. The filters can be dynamically changed to appropriately change the objects being displayed on the map.

(E) Zoom**:** The zoom level of a displayed map identifies its granularity. The zoom level at a client device is chosen based on the user's context. For example, a user inside a museum gets a detailed museum map, but when the user steps outside the museum, he gets an area map of all the museums and other points of interest in the geographic vicinity. The zoom level can be implemented as an attribute of objects, and appropriate filters can then be applied to display a map at the desired zoom level.

(F) Translate: This functionality enables the map service to automatically update the view of the displayed map on the client device as the user moves through the system. When the location of the user moves out of the central region of the currently displayed map, the system prepares a new map display that is appropriately translated from the previously displayed map.



Rover architecture

2.1**. Architecture**: A Rover system, depicted in Figure 1, consists of the following entities

2.1.1 End-users of the system **:**End-users of the system Rover maintains a user profile for each end user, that defines specific interests of the user and is used to customize the content served.

2.1.2 Rover-clients: Rover-clients are the client devices through which users interact with Rover. They are typically small wireless handheld units with great diversity of capabilities in regard to processing, memory and storage, graphics and display, and network interface. Rover maintains a device profile for each device, identifying its capabilities and thus, the functionality available at that device.

2.1.3 Wireless access infrastructure: Wireless access infrastructure provides wireless connectivity to the Rover clients. Possible wireless access technologies include IEEE 802.11 based wireless LANs, Bluetooth and Cellular services. For certain QoS guarantees, additional mechanisms need to be implemented at the access points of these technologies for controlled access to the wireless interface.

2.2. Action model

In order to achieve fine-grained real-time application specific scheduling, the Rover controller is built according to concurrent software architecture we call the action model. In this model, scheduling is done in "atomic" units called actions. An action is a "small" piece of code that does not have any intervening I/O operations. Once an action begins execution, it cannot be pre-empted by another action. Consequently, given a specific server platform, it is easy to accurately bind the execution time of an action. The actions are executed in a controlled manner by an Action Controller. We use the term server operation to refer to a transaction, either client- or administrator-initiated, that interacts with the Rover controller; examples in the museum scenario would be register Device, get Route and locate User. A server operation consists of a sequence (or more precisely, a partial order) of actions interleaved by asynchronous I/O events. Each server operation has exactly one “response handling” action for handling all I/O event responses for the operation; i.e., the action is eligible to execute whenever an I/O response is received.

2.2.1. Actions vs. Threads**:** Our need to scale to very large client populations made us adopt the action model rather than the more traditional thread model. We now provide some experimental justification. There are several ways to use a thread model to implement the Rover controller. One is to implement each server operation as a separate thread. Another is to have a separate thread for each user. Both of these imply a large number of simultaneously active threads as we scale to large user populations, resulting in large overheads for thread switching. A more sensible approach is to create a small set of “operator” threads that execute all operations, for example, one thread for all register Device operations, one for all locate User operations, and so on. Here the thread switching overhead is modest but there are drawbacks. One is that, depending on the threads package, it restricts our ability to optimize thread scheduling, especially as we transit to time based (rather than priority based) scheduling. More importantly, because each operator thread executes its set of operations in sequence, this approach severely limits our ability to optimally schedule the eligible actions within an operation and across operations.

ROVER CLIENT**:** The client devices in Rover are handheld units of varying form factors, ranging from powerful laptops to simple cellular phones. They are categorized by the Rover controller based on attributes identified in the device profiles, such as display properties — screen size and color capabilities, text and graphics capabilities, processing capabilities — ability to handle vector representations and image compression, audio and video delivery capabilities and user interfaces. The Rover controller uses these attributes to provide responses to clients in the most compatible formats. For the wireless interface of client devices, we have currently considered two link layer technologies EEE 802.11 Wireless LAN and Bluetooth. Bluetooth is power efficient and is therefore better at conserving client battery power. According to current standards, it can provide bandwidths of up to 2 Mbps. In contrast, IEEE 802.11 wireless is less power-efficient but is widely deployed and can currently provide bandwidths of up to 11 Mbps. In areas where these high bandwidth alternatives are not available, Rover client devices will use the lower bandwidth air interfaces provided by cellular wireless technologies that use CDMA [11] or TDMA based techniques.

2.4. Rover Controller**:** The interaction of the Rover controller with all other components of the system is presented in Figure 4. The Rover controller interacts with the external world through the following interfaces: • Location Interface: This interface is used by the Rover controller to query the location service about the positions of client devices. The location of a device is defined as a tuple representing the estimate of its position (either absolute or relative to some well-known locations), the accuracy of the estimate, and the time of location measurement. Depending on the technology being used to gain position estimates, The accuracy of the estimate depends on the particulars of the location technology, for example, GPS [6], IEEE 802.11 signal strength, signal propagation delays, etc. Rover takes into account this accuracy information when making location-based decisions.

2.5.Rover Data base:The database in Rover consists of two components, which together decouple client-level information from the content that is served. One component of the database is the user InfoBase, which maintains user and device information of all active users and devices in the system. It contains all client-specific contexts of the users and devices, namely profiles and preferences, client location, and triggers set by the clients. This information changes at a fairly regular rate due to client activities, e.g. the client location alters with movements. The Rover controller has the most updated copy of this information and periodically commits this information to the database. For many of these data items (e.g. client location), the Rover controller lazily updates the database. These are termed as volatile data since any change to these data items are not guaranteed to be accurately reflected by the system across system crashes.

3. Conclusions 3.1. Conclusions and Future Work Rover is currently available as a deployable system using specific technologies, both indoors and outdoors. Our final goal is to provide a completely integrated system that operates under different technologies, and allows a seamless experience of location-aware computing to clients as they move through the system. With this in mind, we have a set of different projects in both the short and the long term.

• Experiment with a wider range of client devices, especially the ones with limited capabilities. They include devices with low-resolution graphics, limited color choices, or only a few lines of text display area.

• For the more-capable devices, we are experimenting with location-aware streaming video services

. • Integrate different other wireless air interfaces to the Rover system. Bluetooth-based LAN is emerging as an important standard today, and it is a logical next technology to experiment with. In the longer term, we are expecting to interact with cellular providers to define and implement mechanisms that will allow Rover clients to interact over the cellular interface.

References

. [1] J. Agre, D. Akenyemi, L. Ji, R. Masuoka, and P. Thakkar. A Layered Architecture for LocationbasedServices in Wireless Ad Hoc Networks. In Proceedings of IEEE Aerospace Conference, March 2002.

[2] P. Bahl and V.N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In Proceedings of Infocom, Tel Aviv, Israel, March 2000.

[3] N. Davies, K. Cheverst, K. Mitchell, and A. Efrat. Using and Determining Location in a ContextSensitive Tour Guide. IEEE Computer, 34(8), August 2000.