Dynamic Composable Computing

Roy Want, Trevor Pering, Shivani Sud and Barbara Rosario
Intel Research
2200 Mission College Blvd
Santa Clara, CA 95024
Email: {roy.want, trevor.pering, shivani.a.sud, barbara.rosario}@intel.com

1. Introduction

In the last 10 years, personal computing has evolved from being primarily a desktop activity to a highly mobile one: the laptop computer, despite its large size and significant weight, has been the most popular mobile platform to date. While smart phones and MIDs (Mobile Internet Devices) have made in-roads into general computing applications, their use is limited to a few key tasks (e.g., calendar, rolodex, mp3 player) that are suited to the small size of their keypad and screen. However, given ever increasing processing and storage capabilities, the potential of these devices far exceeds the computational needs of these applications, and a significant problem facing the mobile industry is how to give users access to a full personal computing experience [9] with the mobility afforded by a smart phone or MID.

This paper proposes a solution to this problem called Dynamic Composable Computing (DCC), which enables the impromptu assembly of a logical computer from the best set of wireless component parts available nearby. Consider the following example which illustrates the goals and flexibility we are trying to achieve with DCC:

> *Fred and Sally are visiting their friend Joe's house when the topic of Sally's recent vacation comes up. Instead of just showing them pictures on her mobile device, Sally displays a collection of her favorite pictures on Joe's wall-mounded flat-screen TV, using her mobile device to advance slides. Meanwhile, Fred takes a moment to browse through Joe's music collection on his mobile device until he finds an appropriate album, which he then triggers to play on Joe's stereo system.*

Our contributions in this paper include: 1) identifying the issues that make dynamic compositions unique, 2) presenting an overview of technology trends that make composition practical today and 3) reporting on the design and experiences of an initial prototype system.

In practice, a mobile device will always be designed based on a compromise that trades-off size, weight, processing power, storage, communication bandwidth, and battery lifetime. DCC aims to overcome these basic design limitations by enabling a platform that is more than the sum of its components: allowing users to easily and seamlessly extend the capabilities of their mobile device with the nearby resources in their environment, and further allow its resources to augment other devices in the locality.

2. Technology Enablers for DCC

There are three emerging technology pillars that support Dynamic Composable Computing: wireless communication, effective processing, and platform sensing.

First, wireless standards provide the ease of creating dynamic connections without requiring a user to physically plug mobile and infrastructural components together. Towards this end, two wireless standards, Ultra-Wideband (UWB)[2] and WiFi-n[14] are now commercially available and enable data transfers up to 480Mbps and 540Mbps, respectively. This improves the throughput of the wireless peripherals making them available at speeds comparable to a wired computer-bus: For the first time we can consider connecting the major system components of a computer architecture using wireless links. Of these two radio options, UWB is likely to be better suited to support composability because the ECMA WiMedia protocol design (Phy & Link Layer) [2] is more power efficient using a timeslot reservation scheme, in combination with a sleep mechanism for unused slots. This is in contrast to WiFi solutions that are based on contention networks, and therefore each node must always be awake and listening to ensure reception of packets.

Second, continuing trends in processor technology are enabling new levels of interoperability between mobile devices and desktop processing ecosystems. Existing low-power processors are improving and even now are powerful enough to effectively run an embedded Linux operating system in a handset; however, they fall short when tasked to run a full desktop suite of applications, including animations, memory-intensive applications, security protection. Furthermore, the general operating environment for mobile devices is different and impoverished when compared with a desktop system, preventing many internet features and plug-ins from operating correctly in a small environment. Solving this problem, a new breed of low-power desktop-compatible processors are entering the market, targeted at MIDs, and expected to bridge the performance gap at low-power, while fully supporting legacy applications.

Lastly, mobile devices are beginning to encompass local sensing to support many alternative forms of interactions. For composition, these sensors are useful for informing a system about the services that are available nearby, e.g. through proximity or physical contact. Mobile sensors are being added to platforms for a variety of reasons, e.g. on digital cameras to determine orientation. Also Near Field Communication (NFC) is a proximity sensing standard that is being integrated with cell phones and small handheld PCs, also called Ultra Mobile PCs (UMPCs), to support financial transactions. Having access to this information, in addition to the results of conventional wireless discovery, allows a device to learn what is nearby, how it is orientated and if it is moving; valuable parameters to enable intuitive connection based on physical attributes that the user can see and directly manipulate.

3. Formative Attributes for DCC

Composability has similarities with conventional distributed systems based on wired networks, for example Plan9 [8] and X-windows [11], which also provide resource sharing solutions. However, a wireless system, unlike a statically connected network, is a dynamic environment in which mobile resources will be serendipitously discovered. This dynamic behavior brings with it a unique set of system challenges relating to how connections are used, their manageability and methods of handling complex multi-device configurations.

3.1 Properties of Wireless Connections

First, wireless computers do not inherently have a link-layer connection with each other and therefore need to explicitly establish a layer-2 connection to use the higher layer-3 protocols commonly employed for service discovery by computers that use wired networks: for example, Universal Plug-n-Play (UPnP). Second, power is a scare resource on mobile devices, and needs to be conserved, thus connections should only form when data needs to be sent, and once formed communication should be minimized. Lastly, wireless communications increases the number of options for users to establish ad-hoc, peer-to-peer connections with the devices in the vicinity. However, this flexibility brings with it cognitive complexity in understanding and keeping track of all the invisible connections between devices, and this problem needs to be overcome to provide the maximum value for users.

3.2 Composition vs. Connection

It is the connection set-up and manageability of a complex system that requires new user-level solutions. However, rather than focus users on individual *connections,* the opportunity for composability is that we can provide users with *composition* commands that operate at a higher level to connect multiple entities together in one logical operation. There are four identified opportunities for composition management: Manual, Named, Task and Context based-operations:

- **Manual:** Users are shown nearby computers and services that they can use, and they are able to *Manually* specify individual connections that are needed to achieve their end goal.
- **Named:** collections of *Manual* compositions saved under a designated name, remembering specific device/service connections. Simply recalling a *Name* allows the user to create a working set of devices with minimal effort.
- **Task**: Specifying a set of resource categories and properties (instead of devices) that are necessary to complete a desired task, utilizing the system to automatically determine which individual devices would be appropriate. For example, if you need a display to watch a movie, the system would suggest the large living room projector instead of a desktop monitor. It differs from *Named* composition, which calls out specific devices and services.
- **Context:** Makes recommendations for compositions based on the current context (everything discovered nearby), effectively suggesting likely compositions based on potential user intent. The system would provide users with a list of task suitable compositions, or named compositions that are now possible given the context. E.g., watching a movie at home in the evening is more likely than during morning hours at work – allowing the system to tailor composition recommendations to minimize user interaction.

3.3 Building on Existing Standards

To enable easy connection set-up between many different kinds of peripherals and keep the system complexity under control, it is necessary to impose a degree of uniformity on the communication protocols needed to link components. The IP protocol is a natural foundation for composability since it supports a wide variety of existing peripheral and network services. Building on the new UWB standard, an IP layer can be achieved through the layer-3 Wireless Link Protocol (WLP) standard, providing peer-to-peer connections over the WiMedia link-layer. By abstracting away device properties with network services, we can seamlessly create multi-device compositions. For instance, common standards for remoting storage and displays are the Samba Protocol, and Virtual Network Computing (VNC) [5], respectively. Using VNC, the keyboard and mouse are also shared as an integral part of service. External broadband network connections such as EDGE or GPRS, can also be shared among computers using IP bridging. However, alternative input technologies such as sensors, for example an accelerometer used to control a local user interface, require a new kind of modular sharing service. Processor sharing at various levels is also possible, but discussion is beyond the scope of this paper.

4. Architecture and Prototype

We have built a prototype composition system that enables users to quickly connect mobile devices together. The prototype architecture is shown in Figure 1, which consists of three components that correspond to these functions: Network Manager, Composition Engine, and the User Interface. Experiences using the prototype are discussed below to frame future work.

Composition Engine (CE): Manages interactions between the User Interface (UI) and the Network Manager (NM), translating the abstract notion of forming a client/service connection into a series of actual commands that execute the connection protocol required to establish that connection. It also maintains a

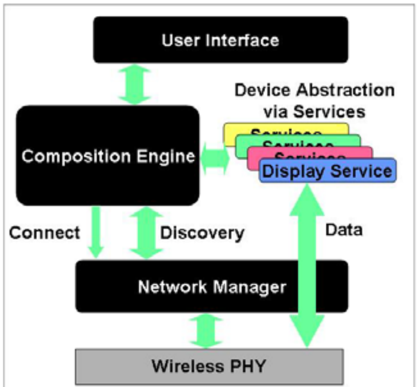

Figure 1: Dynamic Composition Architecture

representation of the state of computers, services and connections; and the security keys required for establishing these connections (provided by a user during the first interaction, but saved thereafter). As resources change on the local machine, it also provides a centralized mechanism for distributing service changes. Finally, it provides a repository of history and profiles for Named, Task, and Context based compositions.

Our prototype CE is implemented in Java to provide a measure of portability between operating systems. Our prototypes needed to run on both Linux and WinXP to make use of the latest UMPCs and MID devices. The majority of our work has been tested on Sony UX50s, a UMPC shipping with WinXP, and is based on an Intel Core Solo U1300 processor clocking at 1.06GHz. The underlying system provides for named compositions that can connect specific instances of devices together.

User Interface (UI): Provides the user with a representation of the computers discovered and their resources available for composition, along with any existing connections. One of the main functions of this component is to provide a composition editing capability, and allow manual connections to be made between clients and servers through a user controlled mechanism. This is also where a user is prompted to provide security credentials for connections. Finally, it provides an interface to specify how a composition could be automated e.g. Named, Task and Context based compositions as described earlier.

Our prototype composition-system user-interface, shown in Figure 2, uses a "join-the-dots" metaphor to create logical computer systems. The center of each cluster represents an individual computer, while the surrounding nodes represent services that a particular device can export. In order to effect a composition, the user can simply draw a line from a service to the desired destination device: active connections are represented by a permanent link between the nodes. This system both allows the user to graphically see what devices and services are available for composition, and also provides an intuitive mechanism to form multi-device compositions, while displaying the entire state of the system.

Network Manager (NM): Abstracts away specific network technologies, allowing the system to learn about changes in network resources as they occur, and at the same time advertise the resources available on the resident computer, and make them available to other nearby computers for their own compositions.

Our prototype system supports a proprietary UWB radio prototype that conforms to the ECMA standard [2] and has a beta-WLP protocol stack available for development. We rely on IP multicast to discover devices over manually connected networks, and the system is able to support any standard underlying network such as Ethernet, 802.11, and UWB.

To provide a perspective on the capability of the UWB radio, a simple bandwidth monitor shows that a 1GB file copy across the link takes ~142 seconds, at an average of 60Mbps. For comparison, a standard 1.5 hour MPG movie can be streamed at the same time as the copy, occupying 2-5Mbps of the link, and the copy will complete in the first few minutes of the movie. While this is impressive it is still not using the full 480Mbps offered by the standard, because these radios are only early prototypes, and the UMPC USB host subsystem is currently a bottleneck in the system.

5. Related Work

There have been examples of systems that provide a framework for various types of composition, but none of them fully support mobile devices as a core component. The iRoom [4] project at Stanford links components together to support work practice in a conference room, using a centralized infrastructure to construct an extended workplace environment. Likewise, Microsoft's EasyLiving [1] project coordinated devices in a home environment, focused on sensing people using vision and interference in a room's infrastructure to decide what connections should be made. Both these systems rely on fixed infrastructure composition, in contrast to the distributed ad hoc approach used in DCC. The Speakeasy [3] project at PARC

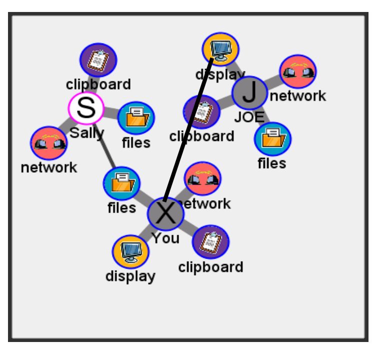

Figure 2: Composition UI; showing a sample composition between three devices. *You* can see *Joe's* display and *Sally* has access to *your* files.

provides for infrastructure network service compositions through mobile code, primarily focusing on fine-grain sources and sinks, such as audio sources and speakers. In contrast, DCC uses legacy device abstractions so that pre-existing applications can automatically make use of these remote devices through *ad hoc wireless connectivity*.

The work on Internet-Suspend-Resume (ISR) [10] at CMU and Intel Research describes how a user's state can be moved from a local to a remote computer by transferring the state of a suspended virtual machine. Although not a direct goal of DCC, it demonstrates how processing can become a composable resource for mobile devices, migrating a user's state from a less capable mobile processor to a nearby high-performance processor in the infrastructure: something DCC can take advantage of in the future.

The Pebbles project [6] from CMU explores using multiple hand-held devices in conjunction with a large-screen public display to allow multiple participants to jointly interact with the shared surface; this project provides a prime example of what can potentially be done with composable systems, but it in itself focuses on specific composed applications and UI generation tools, and does not explore how a flexible platform can support composition, including how device discovery and connection occur.

MobiUS [12] is recent work from MSR allowing two mobile PDA devices to wirelessly combine their screens to show a video that displays across both. This work focuses on the CODEC challenges, rather than generalizing the composition process. Some of our own earlier work on the Personal Server [13] allowed a mobile to view data on a remote screen, but it did not allow building a computer system from an ad hoc set of wireless components.

Proximity interfaces can help in the composition process but lead to ambiguity (e.g., choosing one service out of many possible available). But there are solutions such as "Gesture Connect" [7] which can be used to overcome these issues. All of these techniques can be usefully combined in a general purpose Dynamic Composable Computing system to build wireless compositions as described here.

6. Future Work

The prototype described in the previous section has demonstrated that dynamic composition is a powerful metaphor for mobile computing platforms. However, experience with the prototype has highlighted several areas where further work will greatly enhance the system.

6.1 Ranking Compositions

The current prototype only supports Manual and Named compositions; however, experience has confirmed that they are not sufficient. Manual compositions can be disruptive to a natural workflow since they require many interactions to create a complete composition, and Named compositions are not flexible enough because they are linked to specific devices. In order to provide the best recommendations, we plan to implement Task compositions by ranking all the possible compositions of the wireless devices, and services, found on the wireless network. The users would then be presented with a ranked list of possible compositions; with the highest probability recommendation at the top of the list. A ranking algorithm could be optimized to take into account features, such as minimizing the separation of the component devices, maximize screen area, storage capacity, and link bandwidth.

6.2 Alternative User Interfaces

From the prototype, it is clear that the current graphical service selection mechanism will not scale to environments where large numbers of services are available. Alternative user interfaces may help, such as proximity interfaces provided by the Near Field Communication (NFC) standard, or through the use of spoken commands through a speech interface. NFC has recently been added to our system, enabling devices to connect when brought within 1cm of each other. Speech interfaces have the potential to make composition accessible to a wide variety of users, especially in the home environment. We are keen to see if we can simplify the composition process further using speech alone or in combination with the graphical and NFC approaches we are exploring now.

6.3 Extensible Resource Compositions

One limitation of the current prototype is its ability to support extensible resource sharing. For example, if you connect two VNC display clients running on different machines to a single server, each will currently show the same copy of the server-machine's display. Instead, each could show a different portion of the server screen, enabling two displays to be placed side-by-side creating a much larger viewing area; a distinct advantage for an application such as viewing a multi-column spreadsheet. For every resource category that you may wish to share, it is possible to think of several multi-client use models that could be created by the system. A challenge for DCC is to enable a user to understand the options available, and create meaningful compositions that address the task at hand.

6.4 Layer-2 Service Discovery

In our current prototype the UWB radios are configured to automatically connect to each other to form a local IP network. This solution makes it possible to use conventional layer-3 service discovery mechanisms designed for wired networks, such as UPnP, but it does not take advantage of the low-power potential of the unconnected wireless state when using UWB. However, the services supported by wireless devices are not advertised prior to connection in existing protocols, the default process to find a particular service requires that a wireless node make sequential connections with all other nodes until it is found. This could be a long and tedious process, and for mobile devices reduces battery life.

The next generation of our system prototype will advertise IP services integrated with the layer-2 advertisement and discovery process. In the case of UWB, an advertisement is a periodic information beacon that is transmitted by each wireless device. The UWB discovery protocol allows for data to be encapsulated in fields of a beacon called an Application Specific Information Element (ASIE). Within these fields we can encode the information required for the composition process, as long as it can fit within the assigned data limit (256 bytes), which should be enough to establish a criterion to connect.

7. Conclusion

We believe dynamic service composition based on wireless platforms can be a valuable technique to overcome the constrained resources of small computers. While composition can be a complicated process without the proper systems support, we have demonstrated how system composition can be made intuitive, and presented to users with reduced complexity if the appropriate technologies and standards are carefully woven together. Our initial experiences with composition have yielded encouraging results, highlighting specific techniques to reduce composition complexity.

References

1. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S., "*EasyLiving: Technologies for Intelligent Environments*", Handheld and Ubiquitous Computing (HUC) 2000, Bristol, UK, Sep. 2000, pp97-119.

2. European Computer Manufacturers Association (ECMA) "*High Rate Ultra Wideband PHY and MAC Standard*", ECMA-368 Int. Standards, 1st Edition, Dec 2005.

3. Edwards, K., Newman, M., Sedivy, J., Smith, T., & Izadi, S., "Challenge: Recombinant Computing and the Speakeasy Approach", Proc. of 8th Int. Conf. ACM MobiCom 2002, Atlanta, Georgia, USA, Sept 2002

4. Fox,. A., Johanson, B., Hanrahan, P., Winograd, T., "Integrating Information Appliances into an Interactive Workspace", IEEE Computer Graphics and Applications, Vol. 20. No. 3, May 2000, pp54-65.

5. Li, S. F., Spiteri, M., Bates, J., Hopper, A. "Capturing and Indexing Computer-based Activities with Virtual Network Computing". ACM Sym. Applied Computing, Como, Italy, Mar. 2000, pp601-603.

6. Myers, B. A., "Using Hand-Held Devices and PCs Together," *Communications of the ACM*. Vol. 44, No. 11. Nov. 2001. pp34 - 41.

7. Pering, T., Anokawa, Y., Want, R., "*Gesture Connect: Facilitating Tangible Interaction With a Flick of the Wrist*", Tangible and Embedded Interaction (TEI) 2007, February, 15-17th, Baton Rouge, Louisiana.

8. Presotto, D. et al, "*Plan 9: A Distributed System*", Tech Report. 1991 ATT Bell Laboratories

9. Satyanarayanan, M. "Pervasive Computing: Vision and Challenges", IEEE Personal Comms., Aug 2001, Vol. 8, No. 4, pp10-17.

10. Satyanarayanan, M., Kozuch, M., Helfrich, C., & O'Hallaron, D., "*Towards Seamless Mobility on Pervasive Hardware*, Pervasive & Mobile Computing", Vol. 1, No. 2, June, 2005, pp 157-189.

11. Scheifler, R., Gettys, J., "*The X Window System*", ACM Transactions on Graphics (TOG), 1986.

12. Shen, G., Li,Y., Zhang, Y., "*MobiUS: enable together-viewing video experience across two mobile devices*", ACM Mobile Systems and Applications Conference, MobiSys'07, June 11-14th, 2007, Puerto Rico, USA pp30-42.

13. Want, R., Pering, T. et al. "*Personal Servers: Changing the way we thing about Ubiquitous Computing*", Proc. of Ubicomp 2002, the 4th International Conference, Springer LNCS 2498, Goteborg, Sweden, Sept 30th-Oct 2nd, 2002, pp194-209.

14. WiFi, IEEE 802.11n, October 2007, http://grouper.ieee.org/groups/802/11/index.html