RC24381 (W0710-101) October 18, 2007
Computer Science

IBM Research Report

Celadon: Delivering Business Services to Mobile Users in Public Spaces

S. McFaddin[1], D. Coffman[1], J. H. Han[2], H. K. Jang[2], J. H. Kim[2], J. K. Lee[2], M. C. Lee[2], Y. S. Moon[2], C. Narayanaswami[1], Y. S. Paik[2], J. W. Park[2], D. Soroker[1]

[1]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598
USA

[2]IBM Ubiquitous Computing Laboratory
Seoul
Republic of Korea

IBM

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Celadon: Delivering Business Services to Mobile Users in Public Spaces

S. McFaddin[†], D. Coffman[†], J.H. Han[♦], H.K. Jang[♦], J.H. Kim[♦], J.K Lee[♦],
M.C. Lee[♦], Y.S. Moon[♦], C. Narayanaswami[†], Y.S. Paik[♦], J.W. Park[♦], D. Soroker[†]
[†]IBM T.J. Watson Research Center, Hawthorne, NY 10532
[♦]IBM Ubiquitous Computing Lab, Seoul, Republic of Korea

Abstract. This paper describes the concepts, architecture and implementation of the *Celadon* framework for zone-based services: informational and transactional services offered by businesses to mobile users in public spaces. Celadon is designed to foster an overall ecosystem that addresses the requirements of both the mobile user and the business owner. This is accomplished through an extensible architecture which supports the mobile user through device and context management components, and supports the business owner through classification and reasoning, business process management and interaction management components. The Celadon framework brokers and manages interactions between mobile users and zone-based services which make use of combinations of mobile and facility devices. These interactions take into account preferences provided by the mobile users and policies and processes specified by the business. Celadon enables novel applications that leverage emerging technologies such as REST, data mash-ups, ontologies, and business process modeling. Celadon is currently being used to support pilot implementations of several services in a metropolitan-scale zone based services environment for an experimental "Ubiquitous City" in Korea (the Incheon Free Economic Zone - IFEZ).

1 Introduction

Early proponents of pervasive computing postulated that small, mobile devices and sensors would be commonly associated with humans and would become deeply integrated into their everyday life and business. This concept has enjoyed a first wave of success, especially in wide-area usage domains such voice, text messaging, and email exchange. Wide area usage is characterized by interaction between the mobile user and remote centralized services such as an email or voicemail server. It is widely anticipated [6,18] that another wave of success will be driven by localized usage domains in which the mobile user interacts with rich data and highly functional services provided by the local environment. This will open the door to widespread usage of mobile devices for common functions in the home, workplace, and business domains.

Much work has been done in the area of intelligent pervasive spaces in a variety of settings [24] with a strong emphasis on the user experience. The focus and main contribution of this paper is to address the concerns of both users and business owners. Consider the retail shopping scenario illustrated in Figure 1. The mobile device user, seen at the top of the figure, and the business owner, seen at the bottom, both have a

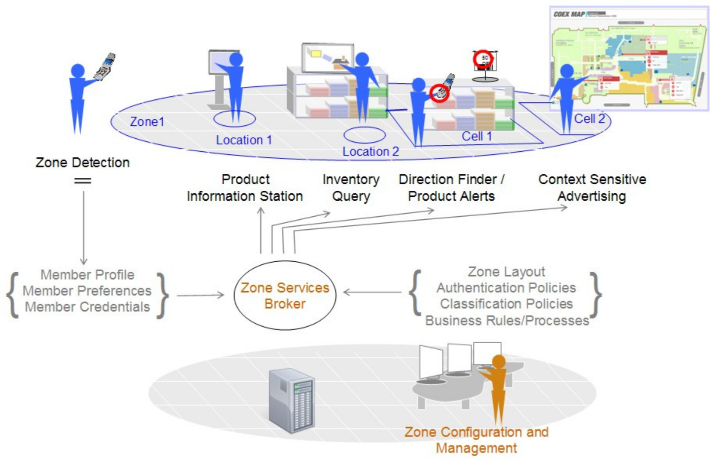

Figure 1: Celadon usage scenario

stake in the successful use of zone based services in this setting. The mobile user desires services to assist him in efficiently performing shopping functions, obtaining useful information, and otherwise enhancing the shopping experience. The business owner wishes to offer services in order to attract mobile device users, guide and influence the user's shopping decisions, and possibly reduce operating costs by providing more self-services to shoppers. We accommodate both sets of wishes by employing a pluggable architecture allowing for easy update of existing services and deployment of new ones.

Our research abstracts the key elements of localized usage domains to a common concept, called zone-based services [23]. Pervasive computing zones (or simply, "zones") generally span a building or a room within a building. A zone based services environment may be comprised of several individual zones. Examples of such public space zones include shopping malls, subway stations, and sports stadiums. Zone-based services are offered by businesses to mobile users in correlation to their physical presence in a zone.

A zone based service is a general access concept, and may reference *web services* – publicly exposed programming interfaces to underlying transactional functions; *data services* – a restricted form of web services offering access to data stores; and *user interaction services* – services offering access to a user interface on mobile or facility devices (such as public displays or kiosks). These types of services may be composed in a symbiotic fashion, involving multiple devices. The concepts of zones and zone based services are applicable to a variety of usage domains.

Individual users of a zone are called *members*. A member is generally *registered* with a zone when the presence of a device owned by the member is detected, or when the member explicitly announces his presence. Members are generally not known a-priori to the zone. During the lifespan of the user's membership in a zone, basic *member context* information is maintained and dynamically updated by the environment. This comprises a battery of information including the user's location, authentication information, profile, preferences and device capabilities. A member's location context is expressed in terms of a hierarchy of three concepts: a *zone*, as noted above, is a logical concept generally corresponding to a room or store; a *cell*, a subdivision of a zone, corresponding to a logical area such as a particular aisle within a store; and a *location*, a point-wise construct, generally corresponding to an individual service point such as an individual shelf.

A zone based services environment constantly monitors the context of its current set of registered members. The environment may make inferences about the condition, intentions, and needs of its members based on information presented by the member at the point of registration. The environment may also respond to dynamically changing aspects of the member context, in particular the member's location. As the member moves around, the environment may suggest or initiate *interactions* which are appropriate for the member. An interaction is a correlation between member and zone services, often involving user input and output on the member's device, a facility device (e.g. a display or kiosk), or both (a symbiotic interaction).

We have constructed a service oriented software framework, called Celadon, for supporting zone based service environments as described above. The rest of this paper describes the requirements, architecture, and implementation of the Celadon framework. It additionally describes a comprehensive scenario we have implemented using the Celadon framework. Celadon is currently being used to support pilot implementations of several services in a metropolitan-scale zone based services environment for an experimental "Ubiquitous City" in Korea (the Incheon Free Economic Zone - IFEZ). Finally, we draw a set of conclusions and lessons learned from our research, and present horizons for further investigation and development.

2 Zone-based Service Scenarios and Design Requirements

We now examine in more detail a retail zone-based service scenario, which we have implemented, to explain the requirements for the design of our system. We envision that users load profile information onto their mobile device or onto a network resident store which acts as a proxy for a device. The user profile could include information with a long life such as customer loyalty cards and payment instruments that need to be guarded safely, transient need-specific data related to the user's immediate shopping goals such as a shopping list ("I need a pair of leather shoes"), or a general set of non-specific preferences ("I like Korean food"). User profile information will be shared with the zone in an ad-hoc fashion as necessary. User profile information is digitally signed during the user's preparatory process using signing authorities recognized by both the device and by a wide set of stores that the user may visit.

Figure 2: a, b) Entering user profiles c) Welcome screen d) Service list e) Location map

The business owner needs to determine the types of anticipated users, the roles they will play, the types of services to be offered, and the underlying business processes which drive those services. A set of rules and policies determine how users are accepted as members of a zone and assigned roles during their membership. Roles characterize the member -- for example the five roles “Shopper”, “Loyalty Program Subscriber”, “Premium Shopper”, “Payer”, and “Employee” serve to distinguish the members. The service analysis considers the physical structure of the environment, and identifies a set of services to be offered at the various zones, cells, and locations as shown in Figure 1. The business process analysis involves classification of members into articulated service levels, and a logical mapping of which services to offer to which classification of members at which location in the zone. The mapping can range from a simple table to powerful modeling constructs such as processes, workflows and business state machines.

As the user enters the zone and embarks upon his shopping expedition a silent bootstrapping process is initiated by physically sensing the presence of the user or his mobile device. The bootstrapping process performs low level cross discovery of device and local services, and in particular identifies core local services against which to register the user. In the member registration process, the device submits appropriate user credentials to the zone (generated and signed above). The environment evaluates and classifies these credentials against a range of policies, starting with low level authentication of signing keys (e.g., is the certificate valid), basic role assignments (e.g., does the certificate represent a loyalty program member), and more advanced classifications (“Premium Shopper”). The bootstrapping process yields a duly registered membership for the user, valid for a specified period of time. A welcome service then provides basic information about the store, and further links to available services and interactions. As the user’s device is equipped with a graphical user interface, it attaches a GUI application to the suggested store welcome service and displays the information (see Figure 2). As the user moves within the store it will offer services that the user can interact with. Generally these interactions are correlated to the user’s location according to the business processes desired by the business owner. The user’s device can dynamically monitor a list of interactions which are suggested by the zone.

As the member moves among the physical cells and locations in the zone, their logical location is tracked, and appropriate interactions are offered in correlation with the location. As illustrated in Figure 1. at “Location1” a generalized product information kiosk is offered (not correlated to any specific product department). At “Location2” an inventory query interaction is offered. This will allow the user to receive highly guided product information tailored to their profile and preferences, their location in the store, and receive information about the availability of the product and direct shipping.

An interaction is made available in “Cell1” area for narrowing product locations (Figure 1) triggered by entry into cell. Product preferences are gathered and correlated to product matches, and the locations are displayed against a small store map, seen on the mobile device.

In some cases a user may want to construct a personalized interaction. For example, the “Cell2” area is dedicated to a form of interaction called a *user driven mashup*. In this type of interaction, an application on the user’s device performs an introspection against the underlying services (e.g. the product information and product inventory services used in the previous interactions). The introspection yields information about the data types exposed by those services (e.g., product information and inventory records). The application can then construct a set of input fields and choices appropriate to that information. These inputs allow the user to construct their own displays of information about the environment and route it to an associated display. The result of such a user-driven mashup is seen in Figure 3 – in this case the user is asking the mashup to search for restaurant listings in nearby zones and plot that information on the large display.

Figure 3: Detailed interaction on large touch screen factoring user preferences – shoe department b) User driven mashup: product location through mobile device and large display symbiosis

By examining the above scenario we can formulate the following design requirements:

(1) Service Brokering. The brokering of relationships between the user and the local environment by computing and managing linkages between services on a device and services in a local zone is essential. Once these linkages are established, existing technologies such as web services can be used to provide rich capabilities for discovering and connecting powerful client applications to fully programmatic services on a network.

(2) Ad Hoc Reasoning. The relationships between a mobile device user and zone based services connections must happen in an ad hoc fashion. In particular, mobile users cannot anticipate each local environment that will enter. Conversely, local environments cannot anticipate the specific users that will enter and make use of local services. Because of this ad hoc nature, zone based services must be based on reasoning and classification. For example, a business may use generalized identity information about a mobile user that has entered a zone to compute service levels (e.g., “Basic Customer” vs. “Premium Customer”) and then use those service level computations to broker appropriate service connections.

(3) Integration of Mobile Device Usage with Business Processes. Businesses need to focus on logical processes rather than information technology artifacts (e.g. databases, customer lists). To make the zone based services concept successful supporting middleware must be amenable to business process analysis and its constituent constructs such as inventories, customer management, etc.

(4) Support for User Defined Interactions. The zone based services concept should offer easy access to rich information in the local environment, and wherever possible, allow the user to construct ad-hoc interactions with the local environment that are not necessarily foreseen and pre-computed.
In addition to these principles, other standard concerns such as security, simplicity and unobtrusive system behavior guided our design.

3 The Celadon Architecture and Implementation

The key challenge in the design of the Celadon architecture has been to relate mobile device processes to business oriented processes. In general, there tends to be a conceptual mismatch between device and business issues – they do not intrinsically relate to each other. Device processes typically focus on low level issues such as making connections, discovering resources, and sensing location. Business processes typically focus on higher level issues such as how to carry out the steps in a multi-part transaction, interact with users, and monitor the overall zone condition.

The design of the Celadon architecture seeks to span the device-business void in a way that builds upon existing technologies and draws them into an overall solution framework for zone based services. Celadon is based on the principles of service oriented architecture. In this approach, each instance of an existing technology is expressed as a s*ervice component*. For example, a device localizer which triangulates devices against 802.11 signal strength would be attached to Celadon as a service component. The external interfaces of Celadon service components have a reduced footprint consisting only of the ability to perform a simple set of data updating operations (their output interface) and receiving a simple set of data events (their input interface).

Service components are gathered into broader architectural layers. For example, the set of service components such as the above triangulator component is gathered into a device management layer. Each layer is driven by a manager component which is responsible for the lifespan of the constituent service components in the layer and for fostering any necessary interconnections among them.

Intercommunication among layers is reduced to a very simple form. Components in different layers do not directly interact, but instead perform operations against a shared set of *data services*. A data service is a reduced footprint web service that models a read/write/monitor data store. Each data service supports a finite set of *data types* and can store an unlimited number of instances of those data types. The interface to each data service is identical and very simple. First, a data service is self describing through the operation *getDataSchemas(),* which returns a listing *{T1, T2, … Tn}* of supported data types. For each supported data type *T* there are four core operations: *addT(t)*, *deleteT(t)*, *updateT(t)* and *findT(t)*. These operations, respectively insert an instance on the store, delete an instance, replace one instance with data from another, and perform a lookup (here the argument *t* is a pattern instance of *T* which found instances must match). The interface also supports event subscription operations *addSubscriber(t)* (*t* is a template instance which serves as an event filter) and *removeSubscriber()* for each data type *T*, yielding events ("*TAdded*()", "*TDeleted*", and "*TUpdated*()").

In this model, a simplified form of intercommunication among the service components happens through the data services. This form of intercommunication requires only that the service components agree to rendezvous on a limited set of data type definitions and their conventions of usage, rather than a complete web services interface. The rendezvous data services are named via a URL and namespace, and data instances are named via the *find()* operation. This makes all data service information globally visible. Service components may then intercommunicate as follows: a component wishing to send a message adds or alters a data instance and a component wishing to receive a message monitors that data instance via the event subscription mechanism.

This form of reduced architectural design, with its limited inter-communication capabilities, is designed to foster adaptability, scalability, and to make the integration of existing software components as simple as possible. New component layers can be added to the design by identifying common data types with which it communicates with other layers. An appropriate layer manager is needed to instantiate and initialize the components in the layer, and to interconnect those components to appropriate data services. The architecture also can be reduced, allowing businesses who do not use various modeling or runtime components to operate using fewer components.

We have used this programming pattern to build an extensive environment for prototyping zone based services applications and scenarios. The full Celadon environment is seen in Figure 4. The central area is the data services bus, supporting both core Celadon data services as well as application data services. This is surrounded by the REST based programming and access layers, which provide facilities for establishing the linkages described in Figure 4. At the top are a set of component layers which drive the runtime activities of the zone. These component layers intercooperate, for example, to enable the types of end-to-end scenarios described in the previous section. Note that the activity layers range from the device oriented activities at the left, to business oriented ones at the right. The activity layers have been arranged in this order to describe a general programming pattern which begins with device activities, using data rendezvous in the middle, and rippling through the business process reasoning in the middle, culminating in business interactions at the right. The component layers at the bottom are used for configuration of the zone and for the runtime management and monitoring. These layers may be augmented and swapped out in various deployments – for example if a business owner does not make use of classification and reasoning technologies, the overall environment still operates.

The following sections describe each layer at a summary level, and will illustrate the end-to-end operation of the Celadon architecture in managing the scenario previously described. Additional papers in preparation describe in further detail the Celadon programming model, as well as the operation of the specific service component layers.

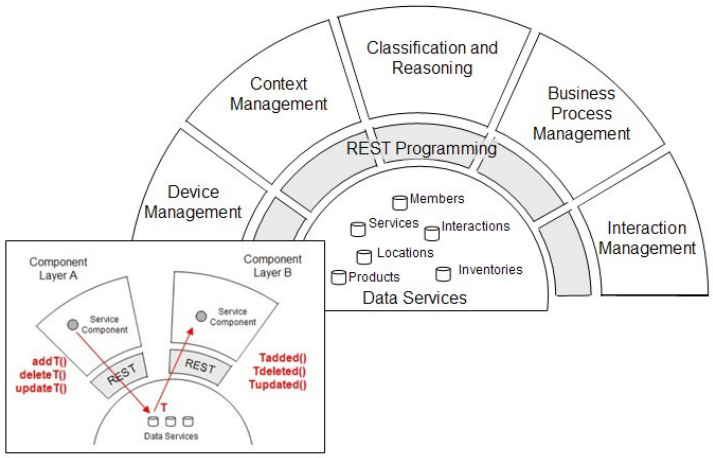

Figure 4: The architecture uses a repetition of the basic pattern illustrated in the inset**.**

3.1 Data Services Layer

The data services layer, seen in the center of the architecture, provides a shared, addressable information space for the environment. The information space captures the core artifacts of the environment, including the zones, cells, and locations within the environment, the services offered in the environment, the members currently registered with zones in the environment, logical and physical location information about the members and services, and the status of interactions in the environment. In our prototype implementation, the data services layer is composed of the following:

Member Information Service. This is a core Celadon data service which carries a set of *member information records*, describing each registered member and carrying the member's credentials, preferences, and logical roles assigned to the member.

Service Information Service. This is a core Celadon data service which carries a set of *service information records*, describing each service available in the Celadon environment. Service information records provide a rich data description of the service, a URI for the service, and a status code indicating if the service is active, or available.

Location Data Service. This is a core Celadon data service which carries a set of *location information records* describing the physical and logical location of zone members. This information is updated by various location brokering components as the member moves around within the environment and may be monitored to trigger service and interaction offerings in correlation to the member's location.

Interaction Information Service. This is a core Celadon data service which carries a set of *interaction information records*, describing interactions that are available to zone members. An interaction correlates one or more services into a logical application where each service provides a constituent part (display service, data service, etc.)

Product Information Service. This is an application data service which carries a set of *product information records*, describing the technical details of a product offered in a retail setting.

Perpetual Inventory Service. This is an application data service which carries a set of *perpetual inventory records*, describing a quantity of a particular product correlated to a logical location.

3.2 Device Management Layer

Service components in this layer handle the incorporation of devices into the local environment. Device management services may be specific to the physical configuration of the local environment and to specific device platforms, whereas the components in other service layers are not. Our prototype Celadon environment provides several device management components which support a LAN-oriented environment with PDA class devices.

Typical categories of device management components include:

Presence Detection Components. Presence detection components sense the presence of members in a zone and may be either automatic, allowing the zone environment to be recognized as a background activity, or gestural, requiring explicit actions from the zone member. In our LAN-oriented implementation we provide a multicast DNS based presence detection component which uses "Zero Configuration" 4 network sensing. Devices interoperate with presence detection components to carry out a bootstrapping process where the core Celadon services are identified, and the user is registered as a member via credential submission.

Location Detection Components. These types of components handle low level sensing operations which report location information about zone members.. In our LAN-oriented implementation we have provided a simple location detection broker which correlates 802.11 radio signals to the physical (X,Y,Z) coordinates and reports that to the location data service. Other location detection approaches are commonly available, including passive methods (e.g involving RFID tags carried by members), and active methods requiring users gestures (e.g. near field communication swipes) at service points.

3.3 Context Management Layer

Service components in this layer are responsible for maintaining logical-level context information about the zone environment. Context management components are generally unaware of the physical configuration of the zone or of specific device platforms in use. In general, they monitor the lower level physical information written into the data services layer by the device management layer and upgrade that information to a logical level. Key categories of context management components include:

Location Brokers. This category of context management component is responsible for conversion of location information from its physical forms into logical forms. Location brokers work with a higher level description of the zone environment such as a planogram and convert low level location data (e.g. *{X,Y,Z}*) into higher level geometric constructs which yield logical level location correlations (*{ZoneID,CellID,LocationID}*) for zone members.

Credential Brokers. This category of context management component is responsible for conversion of low level member credentials into logical roles. Credentials (certificates, digitally signed documents, etc) are received at the time of the initial registration of the member with the zone environment. The context management layer is populated with a battery of credential brokers, each designed to handle a particular type of credential anticipated from the zone users. Credential brokers inspect each appropriate member credential, validate it (check the signing data), and decompose it into information about the signer, and the content which was signed. The credential broker then applies a policy which determines what role the bearer may subsequently play in the environment. We have implemented credential brokers based on two types of public key infrastructure (PKI) technology: X.509 public key certificates, and XML digital signature documents. This scheme provides for a simple but reliable way of receiving un-administered members into a zone. This is an important characteristic of public space systems, because it allows any user to walk up and use the zone and its services, so long as they carry credentials signed by a trusted authority.

3.4 Classification and Reasoning Layer

The profusion of data of an ad hoc nature on the world-wide-web has led to the development of new classes of sophisticated tools for describing data semantics. These tools allow data to be self-describing and therefore much more easily discoverable, maintainable and useable. Data, once appropriately classified, may be used as the basis for a formalism of reasoning in which relationships among various classes of data may be defined and inferences may be formed thereon.

Celadon faces a similar situation in the multiplicity of members, services and other artifacts found within a zone based service environment. The Celadon framework makes use of the semantic modeling tools for the classification of these items in its Classification and Reasoning Layer. Components in this layer extend the basic capabilities of the context management layer to enact deeper levels of inference and reasoning about the context management artifacts (members, services, interactions, etc) – in particular dealing with the concepts of purpose, role and intent. These inferences can subsequently be used to enact a business process or trigger services that provide customized treatments, for example, a distinguished service level for a zone member. The classification and reasoning methodology employed by these services are typically driven by policies, rules, or ontologies. As with the other service component layers, the classification and reasoning layer communicates indirectly with the other layers, monitoring the data services layer, responding to updates, applying reasoning and inferences to those updates, and writing inferred knowledge back into the appropriate data services layer.

As an example of the use of reasoning, the Celadon implementation computes and assigns member roles when the user enters a zone. Member roles are represented as an instance, or *individual*, of an ontology of roles. Ontologies in Celadon are composed in the OWL language. OWL expressions capture the notion of classifications -- for example, a Celadon member is a kind of person, a shopper is a kind of member, and a premium shopper is a kind of shopper. OWL also represents restrictions on roles: a manager must have the privilege of being a employee, and further, must manage a business that is itself a subscriber to the Celadon zone.

The computation of roles, as indeed, all of the reasoning accomplished by this layer, proceeds by comparing the information available from the context layers with an ontology. Most of the comparisons are accomplished through the use of a reasoning engine [13]. However, as is well-known, certain calculations, such as numeric comparisons, cannot efficiently be carried out by a reasoning engine. Therefore, some of the computations are addressed to a procedural engine.

The same approach can be applied to other context artifacts maintained by Celadon. For example, ontologies can be used to reason about user interests. If a user indicates an interest in a particular movie, for example, an ontology of movies can be used to reason that the user would be interested in another movie starring the same actors. Similarly, if the user has expressed a desire to receive discount coupons for clothing sales, an application with knowledge of the types of goods available in local stores might offer the user a coupon upon approaches a clothing store.

3.5 Business Process Management Layer

Recent years have witnessed a transformation in business information technology towards a concept called *business process management (BPM)*. In this concept, a business recognizes and captures its high level business processes and practices as core information technology assets, in addition to traditional assets such as databases and servers.

An increasing body of technology is available which allow business owners to model the operation of a business at this level, providing greater adaptability to changes in their business models by making them independent of lower level information technology details. Celadon supports a business owner in incorporating this technology into zone based services environments through the business process management layer.

Two key constructs from the BPM field are particularly applicable to zone based services environments. *Workflows* model patterns of activities that consist of sequences, decisions, loops and similar constructs. This is useful in zone based services environments to model interactions in which users are guided through well defined sequences of steps and branches. *Business state machines* model event driven activities which transition among a set of well defined states. This is useful in modeling human driven activities in which the system cannot necessarily prescribe each step, and must instead rely upon human volition.

The business process management layer provides a set of services that initiate and manage the high level business processes of a zone. Each service in this layer is responsible for a business process management (BPM) construct such as a workflow, decision table, or business state machine. The service is responsible for the instantiation and initialization of the construct, feeding it parameters, and subsequently directing and marshalling inputs as events occur in the underlying data space.

For example, in our Celadon implementation we have used business state machines which combine the logical role information associated with a zone member with logical location information to initiate customized, location-sensitive services and interactions. State machine instances are defined with logical location constructs such as "Cell2" modeled as states and user activities such as location changes modeled as state transitions.

Such models are constructed by defining state transition diagrams using high level modeling tools (e.g. visual state diagram editors), compiling the diagram into executable code, and mounting that code as a service element in the business process management layer. The resulting state machine component can be bound to artifacts in the data services layer, for example, to handle a zone member entering and exiting a logical cell location. Specific actions are programmed against procedural hooks called partner links which are bound to the state diagram..

Partner links are associated either with state transitions or state arrivals. For example, one partner link could be a simple action that creates an interaction record and adds it to the interaction information data service. This, in effect, would cause a specific type of interaction to be either suggested to the user or automatically initiated upon cell entry or exit. For example, an interactive advertising interaction could be programmed to start with the cell is entered. Similar binding approaches may be used for other constructs such as business process diagrams ("workflows").

3.6 Interaction Management Layer

User interaction with a Celadon zone is done via personal mobile devices or through public facility devices, such as large displays and kiosks, and may involve multiple devices interacting symbiotically. Celadon models interactions as logical combinations of display, application, and data services. For example, the screens seen in Figure 2 are each a combination of an on-device display service with an underlying data service. The screens seen in Figure 3 are combinations of facility display services (the large format display in 3a and 3b), device display services (3b inset) and data services (e.g., the product information data service).

The interaction management layer supports a set of interaction service components that control access to the interaction elements provided by the zone environment – for example, each facility display is represented by such an interaction service component. The service component monitors requests to access the interaction point and imposes access policies. For example, there may be multiple individual zone members attempting to initiate interactions with a single facility display.

Each interaction is represented by an interaction information record and is maintained by the interaction information service. An interaction has a lifecycle, indicated by a status field, with example values "available" (a suggested interaction) and "active" (an in-process interaction). Interactions are correlated to zone members, and during a single zone visit, a given member may have a wide variety of interactions made available (and possibly withdrawn).

Interactions are typically suggested by business process constructs, which enter new interaction information records into the interaction data service. These interactions may be subsequently initiated either automatically or by explicit user choices. Automatically initiated interactions may be used in cases where user confirmation is not desired -- for example, a context sensitive advertising interaction may be started as soon as a zone member walks by. User initiated interactions may be managed by on-device applications. Figure 2d illustrates an interaction composed of an on-device service chooser (the display service) with the interaction information service itself (the data service). The display is kept up to date by monitoring interaction information record events, filtering against the particular member carrying the device. When the user selects a particular interaction, the service chooser activates the underlying interaction record.

There are two key design points to this approach. First, the use of the interaction data service functions as an intermediary which allows the incorporation of business process logic into the decision making process about when and where interactions are important. Second, it provides a service composition approach to expressing interactions – this enables symbioses of multiple display and service elements.

We have used this design approach to enable a particularly advanced example of interactions: user driven mashups. These interactions are compositions of multiple display and data services which yield hybrid information displays. These types of interactions overlay information from one service onto another by correlation of interior data elements. For example, product location data from a product information service would be correlated to location data from a planogram service (e.g. providing a store or mall layout) to overlay products on a store map. We have used this approach to construct the product/service finder mashup seen in Figure 3. The data sources need not be predetermined and may be chosen or parameterized by the user. For example, as seen in the inset, the user is constructing a query on the mobile device screen (a restaurant query) which filters the information being channeled into the larger mashup screen, resulting in a product map of the zone member's favorite restaurants.

4 Related Work

Delivery of diverse services through mobile devices is increasingly becoming a reality [18,6]. To date, a number of zone-based services have been built and have enjoyed some success, but for specific functional offerings and in specific geographies. Some examples include: Korea Telecom (KT) offers contact-less payment in subways and coffee shops. A ubiquitous society experiment was conducted in Tokyo's Ginza district early 2007 [11]; Disney's Pal Mickey is a wireless toy that provides contextual information and alerts for nearby events at Disney World in Florida [8].

Our work draws upon and extends a wide body of previous research. Active spaces research has explored the integration of humans with sensor-rich environments. The GAIA project at UIUC has focused on classroom and similar environments [22]. The "Aware Home" research initiative at Georgia Tech has focused on a smart home environment 1. The Stanford iRoom project focused on services necessary for group collaboration in a research environment [13]. The MIT Project Oxygen [19] and CMU Aura 10 focused on the user and user attention, studying how pervasive computing systems can serve us rather than making the system be the focus. The CHIL project, funded by the EU, has similar goals 4. Similarly, the Portalano project from U. Washington focused on making user tasks a focus of pervasive computing [9], and the related Place Lab project at Intel Research has studied the use of low-cost device positioning techniques for delivering location-aware applications [12]. The Context

Sensitive Tour Guide [7] provides location specific information about tourist attractions. BlueSpace, from IBM, has focused on the office workspace, creating smart cubicles [16]. The Microsoft Easy Living effort [3] is concerned with developing an architecture and technologies for intelligent environments which allow dynamic aggregation of diverse I/O devices into a single coherent user experience, with focus on home and office environments. HP CoolTown is an infrastructure to support web presence for people, places and things [15], where virtual and physical elements are associated. Although the above efforts relate to diverse contexts of smart spaces, we believe our research breaks new ground in building environments in the public business setting, focusing on the needs of both end users and business owners.

Mobile web services have focused on the infrastructural issues related to bringing the web services paradigm to mobile and pervasive computing [2]. This has served to upgrade the basic computing environment for pervasive computing to that available for desktop and server programming. Business process management (BPM) focuses on converting transactional business systems to a model-driven approach, and has been extended to the pervasive space [21] -- here we augment the treatment for public zones.

5 Conclusions, Status and Future Work

This paper has presented the Celadon framework for connecting mobile users to business services in public settings. The paper outlines an abstract model for such environments, called zone based services, and describes a pluggable service oriented architecture which enables business owners to rapidly and efficiently deploy these scenarios. The architecture is broadly conceived to accommodate the requirements of both mobile users and business owners.

We have created a substantial implementation of the Celadon architecture and have applied it to a realistic usage scenario involving LAN connected, PDA-class devices in a zone based services environment with multiple interaction points. Additionally, a new information system featuring the Celadon infrastructure is being installed in the Incheon Free Economic Zone (IFEZ) near Seoul, Korea. IFEZ comprises three new "Ubiquitous Cities" with facilities for both living and working, and an emphasis on advanced information technology. The Celadon installation provides services of three major types: information mashups for zone visitors on large format (60 inch displays), Ubiquitous-community (U-community) services, such as language learning, for zone employees, and a service for managing conference rooms, controlling projectors, lighting, projector screens and presentations via smart phones.

Much additional research remains. The architectural approach uses a hub-and-spoke design having a set of REST-based data services at its center, with an eye toward simplicity and ease of extension. This allows the conceptual decomposition of a zone based services domain into a small number of data types describing the core artifacts of a zone. Research is needed as to whether this can be used in the long term as a strategy for modeling complex zones. Also, note that the interconnection among service components, using the central data services layer as a rendezvous point, forms a defacto programming model. Possible future work would be to formalize this as a model for wiring and programming zone based service environments. Additional research is needed into the performance of such a design as deployments scale to large numbers of users or complex zones. Research is also needed into the extent to which readily available constructs from the business modeling sphere suffice to support rich user activities. Using ontologies to infer user intent is also a rich research area.

References

1. G. Abowd et. al., "The Aware Home: Developing Technologies for Successful Aging", *proc AAAI-02 Workshop "Automation as Caregiver"*, pp. 1-7, 2002.
2. S. Berger et. al., "Web Services on Mobile Devices - Implementation and Experience", *5th WMCSA*, pp. 100-109, 2003.
3. B. Brumitt et. al., "EasyLiving: Technologies for Intelligent Environments", *Proc. 2nd Intl. Symp. Handheld and Ubiquitous Computing*, pp. 12-29, September 2000.
4. S. Cheshire and D. Steinberg, "Zero Configuration Networking", O'Reilly 2006.
5. CHIL: Computers in the Human Interaction Loop: http://chil.server.de/servlet/is/101/
6. Constantiou, J. Damsgaard & L. Knutsen, "The Four Incremental Step Toward Advanced Mobile Service Adoption", *CACM*, pp. 51-55, June 2007.
7. N. Davies, et. al., "Using and Determining Location in a Context-Sensitive Tour Guide," *IEEE Computer*, 34(8): pp.35-41, 2001.
8. Disney Pal Mickey: http://en.wikipedia.org/wiki/Pal_Mickey
9. M. Esler et. al., "Next century challenges: data-centric networking for invisible computing", *5th MOBICOM*, pp. 256-262, 1999.
10. D. Garlan et. al., "Project Aura: Toward Distraction-Free Pervasive Computing",*IEEE Pervasive Computing*, pp. 22-31, April-June 2002
11. Ginza: http://www.tokyo-ubinavi.jp/en/ginza.html
12. J. Hightower, A. LaMarca and I.Smith, "Practical Lessons from Place Lab", IEEE *Pervasive Computing*, 5(3), pp. 32-39, 2006.
13. IBM Scalable Ontology Repository: http://www.alphaworks.ibm.com/tech/semanticstk/
14. B. Johanson & A. Fox. "Extending Tuplespaces For Coordination in Interactive Workspaces", *Journal of Systems and Software*, 69(3), pp. 243-266, Jan. 2004.
15. T. Kindberg et. al., "People, Places, Things: Web Presence for the Real World", *Mobile Networks and Applications*, 7(5), pp. 365-376, October 2002.
16. J. Lai et. al., "BlueSpace: Personalizing Workspace through Awareness and Adaptability", *International Journal of Human-Computer Studies*, 57(5), pp. 415-428, 2002.
17. S. McFaddin, et. al., "Modeling and Managing Pervasive Computing Spaces using RESTful Data Services", submitted, 2007.
18. M. Megna, "Mobile Commerce: Tapping the Untethered Market", *Ecommerce-Guide.Com*, February, 2007.
19. MIT Project Oxygen: http://oxygen.csail.mit.edu/
20. O'Connell, P.L., "Korea's High-Tech Utopia, Where Everything Is Observed", *NY Times,* Oct 5 2005
21. A. Ranganathan and S. McFaddin, "Using Workflows to Coordinate Web Services in Pervasive Computing Environments", *2nd Intl Conf. on Web Services* pp. 288-295, 2004.
22. M. Román et. al., "Gaia: A Middleware Infrastructure to Enable Active Spaces", *IEEE Pervasive Computing*, pp. 74-83, Oct-Dec 2002.
23. M. C Rosu, et al., "Celadon: A Novel Architecture for Symbiotic Computing," In *Proc. International Symposium on Ubiquitous Computing Systems* (UCS) 2006.
24. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, 8(4), pp. 10-17, Aug. 2001.