AmbiTalk: Enhancing Wireless Communication Services through the Automatic Adaptation of Mobile Communication

Eric Karmouch, Patrick Martin, Hossam Hassanein

Abstract—**AmbiTalk is a ubiquitous system based on the Session Initiation Protocol (SIP) and Bluetooth that allows a mobile device to automatically adapt the behavior of its communication services as its user moves from one location to another. The adaptation is policy-based and occurs both pre-call and mid-call. We present the AmbiTalk architecture and discuss how it allows a device to automatically adapt its communication properties to its current environment and other devices. We demonstrate the viability of the AmbiTalk approach with the implementation of a prototype.**

***Index Terms*— Ubiquitous location-based services, context-aware computing, Internet multimedia, mobile communications**

I. INTRODUCTION

As advancements in IP-based multimedia communications continue, mobile devices become increasingly embedded into daily life, accompanying and actively participating in a person's daily routine. A mobile user traveling from one location to another requires their device to adapt behavior. For example, upon entering a meeting room, a user might manually tailor their device to a more discrete state, possibly opting to alter their device's alert setting to a vibrate function and switch their device's communication mode from voice to instant messaging. Alternatively, if entering a meeting room while currently in an active call, a user might choose to end their voice call and begin an instant messaging call on the same device. The decision as to how to tailor the behavior of a device reflects a user's view of a location's policies. As the policies of a location are not always known or respected [1] and continuously adapting the behavior of a mobile device can prove tedious, the ability for a mobile device to adapt its own behavior becomes an asset.

To provide this ubiquitous service, we build a system incorporating context-awareness into device-mediated communication. We refer to this system as AmbiTalk, due to its contribution to the Ambient Intelligence paradigm [2]. Moreover, we develop various AmbiTalk solutions, namely the Blue-Poll Scheme (BPS), the Ubiquitous Interaction Model (UIM), and the SIP Session Adaptation Mechanism (SSAM), providing intelligent behavior reasoning and adaptation.

The system is based on the Session Initiation Protocol (SIP) [3] to benefit from three of its advantages: 1) the increasing deployment of SIP-enable software and hardware clients, 2) the Third Generation Partnership Project (3GPP) adopting SIP as its standard for session establishment in the IP Multimedia Subsystem (IMS) [4], and 3) SIP being considered a feasible application layer approach to mobility management [5]. Furthermore, the system includes the use of Bluetooth [6], as Bluetooth capabilities have become a standard functionality in mobile devices.

In this paper, we present the AmbiTalk architecture and describe our prototype. In Section II, we provide related work in the area of automatic behavior adaptation and SIP-based context exchange. In Section III, we provide an introduction to SIP. In Section IV, we discuss system requirements and design. In Sections V and VI, we present the details of the architecture, and in section VII, we describe our prototype system. We conclude in Section VIII by discussing future work.

II. RELATED WORK

A number of existing approaches address context-based automatic adaptation in mobile devices. An approach of interest is that of Mika Raento et al's ContextPhone [7]. This solution directly senses context from the environment through the use of sensors embedded within the mobile device. Behavior adaptation is completely localized to the device and is a result of data acquired by a device's physical and virtual sensors. Siewiorek et al's Sensay project [8] presents a similar technique. These approaches focus on the context-based adaptation of mobile device behavior, enhancing the user experience. However, they do not provide the means for a location to possess and enforce policies. In contrast, the solution of Kay Connelly et al [9] takes location policies into consideration, as it proposes a framework for the automatic configuration of mobile devices within smart environments. Our solution differs, as we extend our approach to handle the

Eric Karmouch is with the School of Computing, Queen's University, Kingston, ON K7L 3N6 Canada. (e-mail: karmouch@cs.queensu.ca).

Patrick Martin is with the School of Computing, Queen's University, Kingston, ON K7L 3N6 Canada. (e-mail: martin@cs.queensu.ca).

Hossam Hassanein is with the School of Computing, Queen's University, Kingston, ON K7L 3N6 Canada. (e-mail:hossam@cs.queensu.ca).

adaptation of active media sessions.

A SIP-based technique for exchanging context is given by Manuel Gortz et al [10]. This approach acquires communication context provided by the environment and proposes a means of sharing this context between a caller and a callee through the enhancement of SIP. The goal of this technique is to provide knowledge of the callee's context to the caller, leaving the caller to manually interpret the context and adapt the behavior of their device. Our approach differs, as it incorporates interpretation and adaptation as part of the automatic process.

III. Background

The AmbiTalk architecture extends the existing SIP architecture. This section introduces SIP and its architecture.

A. *Session Initiation Protocol (SIP)*

SIP is a textual client-server protocol [3], where the client issues a request and the server returns a response. SIP provides the means to establish, maintain, and terminate IP multimedia sessions. SIP does not transport the media itself, but rather provides the signaling and media description necessary for another protocol (e.g. RTP [11]) to handle the task. SIP is composed of user agents and network servers. User agents take on the roles of *User Agent Client* (UAC) or *User Agent Server* (UAS), and network servers take on the roles of proxy, redirect, or registrar. User agents represent the user, whereby UACs initiate a SIP request and UASs receive a SIP request and return a SIP response. The communication between user agents is accomplished either directly or through the use of SIP network servers (i.e. proxy, redirect, and registrar). *SIP proxy servers* provide the correct forwarding of incoming SIP requests by receiving a SIP request, determining the next hop, and forwarding the request. A *SIP redirect server's* only task is to receive a request and instruct the client to contact the next hop directly. *SIP registrars* are key components of SIP, such that they act as the registration authorities within SIP domains and are responsible for user location management.

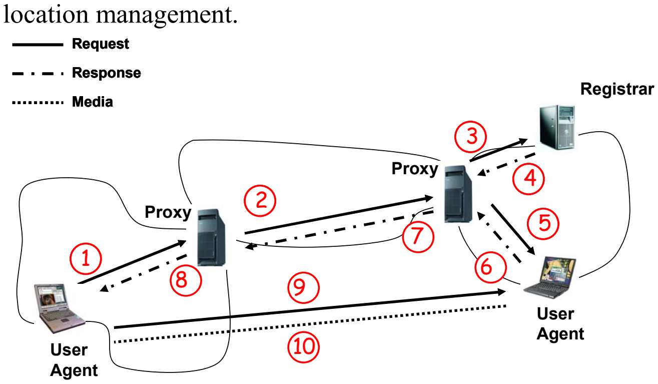

Fig. 1. Example SIP message flow

SIP request and response messages contain a number of headers and consist of several methods such as INVITE, BYE, 200 OK, and ACK. These methods are used to specify the actions required by the participants. Figure 1 depicts an example sequence of messages used in a media session establishment. A general session initiation requires a UAC to send an INVITE request (Action 1), which is successfully acknowledged by a UAS through the response of a 200 OK message (Action 6). When a UAC receives a 200 OK message, it replies with an ACK message (Action 9), finalizing the session initiation. A session is terminated through a BYE request, which is also acknowledged by a 200 OK response.

IV. System Requirements and Design

The system architecture presented in this paper is divided into two concepts and is guided by a set of requirements. We begin this section by identify these concepts and requirements.

A. *Design concepts and requirements*

Enabling a device to interpret the policies of a location and adapt its behavior from one location to another, leads our architecture to consist of two main design concepts: *policy discovery/reasoning* and *communication adaptation*. Policy discovery/reasoning implies a device's continuous awareness of the policies of its location and the behaviors it can be tailored to in order to meet those policies. Communication adaptation implies a device's ability to adapt the behavior of its communication services, based on the policies of its location, both pre-call and mid-call. Furthermore, we require our system to be:

- *Light weight:* No physical sensors are required by the mobile device. A location communicates a simple conclusive context representing policy reasoned from gathered environmental context.
- *Flexible:* Multiple behaviors are able to satisfy a location policy, such that the behavior of a device is interpreted from policy and is negotiable.
- *Automatic:* A user has a minimal role in the process of policy discovery/reasoning and communication adaptation. The role of a user only extends to the predefining of preferences and the response to solicitation for confirmation.
- *Seamless:* From a user perspective, communication adaptation is completely transparent when performed pre-call and as seamless as possible when performed mid-call. Mid-call adaptation requires media disruption, but does not appear to the user as a new call.

B. *Architecture overview*

The AmbiTalk architecture (Figure 2) is designed to extend and compliment the existing SIP architecture, by replacing SIP's traditional user agent with the *AmbiTalk User Agent* and including a new agent called the *AmbiTalk Location Agent*. The rest of the SIP architecture (i.e. proxy server, redirect server, and registrar server) remains unchanged.

The AmbiTalk user agent (AT-UA) is an enhanced version of the SIP user agent, such that it is capable of providing all the functionalities of the traditional SIP user agent in addition to developed AmbiTalk functionalities for behavior adaptation. Figure 2 depicts the functional organization of the

AT-UA. The AT-UA is made up of four component functions, namely the *SIP User Agent* (SIP UA) function, the *Service Functionality Controller* (SFC) function, the *micro Context-Aware System* (mCAS) function, and the *Controller* function. The SIP UA function provides the execution of the Session Initiation Protocol, the SFC function provides the control of communication services, the mCAS function performs behavior reasoning, and the controller function guides the actions of the AT-UA. As Figure 2 depicts, the AT-UA is located on a user's physical device.

The AmbiTalk Location Agent (AT-LA), located on the Bluetooth server, is the second AmbiTalk agent and has only one *Main* function; to dispense location-based policies to the AT-UA. Therefore, any location wishing to enforce its policies is required to contain the AT-LA. The AmbiTalk system envisions the AT-LA as being integrated into a *context-aware environment*, such that the policies dispensed by the AT-LA are composed from a *context-aware system* (CAS), utilizing physical sensors to collect and reason real-time information from the environment. Section V.A describes the ideal CAS for such a task.

The functionalities of the AmbiTalk system are achieved through the interaction of AmbiTalk agents (i.e. AT-UA, AT-LA). The AT-UA interacts with the AT-LA using the AmbiTalk developed *Blue-Poll Scheme* (BPS), described in Section V.C. The AT-UA interacts with other AT-UAs using traditional SIP requests and responses injected with an additional *AmbiTalk header*, described in Section VI.B. All agent interaction is bilateral and follows the strict AmbiTalk developed *Ubiquitous Interaction Model* (UIM), described in Section V.E. Furthermore, the AT-UA ensures that all communication services involved in a media session are negotiated and meet the policies of its respective location. If the policies of a media session change mid-call the AT-UA is able to adapt communication in real-time using the AmbiTalk developed *SIP Session Adaptation Mechanism* (SSAM), described in Section VI.C.

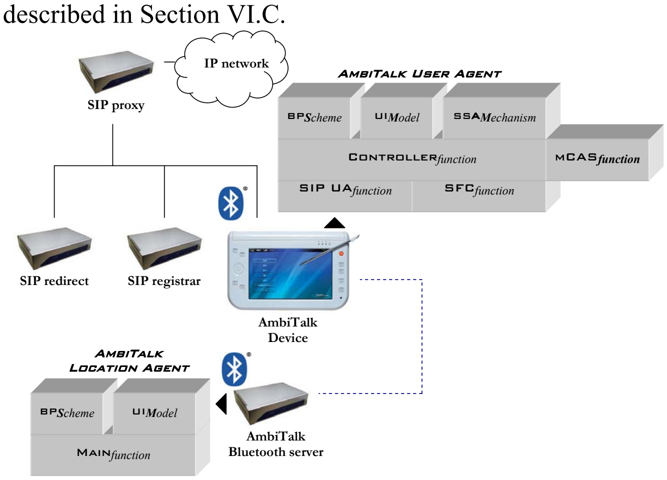

Fig. 2. AmbiTalk architecture

V. Policy discovery/reasoning architecture

The AmbiTalk architecture envisions an existing context-aware system (CAS) as being present in the location. This section begins with a discussion of the ideal CAS infrastructure, leading to the defining of AmbiTalk policies and the presentation of the workings of the AmbiTalk architecture in discovering location policies, reasoning behavior through policy proposals, and realizing a model for ubiquitous bilateral interaction.

A. *The ideal context-aware system infrastructure*

The acquisition and interpretation of contextual information is defined as *context-awareness*. Context is "any information that can be used to characterize the situation of a subject and its interaction with optional objects." [12] Gathering contextual information requires a *context-aware system*. Context-aware systems contain the appropriate infrastructure to allow for three basic actions, namely *context acquisition*, *context representation*, and *context reasoning*. There are many context-aware systems in the research; a survey is provided by Baldauf et al [13]. Context acquisition requires sensors, physical or virtual, used to sense signals from the environment. Sensed signals can include temperature, acceleration, location, or any other signal that contributes to the modeling of the real world. Sensors sense data in its raw form. Raw data has little meaningful value until it is translated into a proper context representation (e.g. relevant units). Once translated, context can be analyzed and aggregated to form real world properties. The analysis and aggregation of context is based on context models. There are a number of approaches to context modeling; a survey is provided by Strang et al [14]. An advantage of using context models is the ability to directly extract *policies* from the model.

B. *Defining policies in the AmbiTalk system*

Typically, policies are expressed as deontic modalities on actions such as permissions, obligations, and prohibitions [15], and act as a prescriptive means of restricting a system's behavior throughout different situations; a concept known as *policy-awareness* [16]. These deontic modalities can be formed in terms of 'if-then' condition statements represented as a form of predicate logic. For example, a policy obliging a motion sensor to turn on a set of lights can be expressed as, *"If and only if the sensor for lightX is true, then activate lightX"*

```
Activate_Lights(lightX) iff
    Sensor(lightX) == true
```

The AmbiTalk system integrates this format for policy representation and extends it to include *nested policies*, *explicit policies*, and *implicit policies*.

Policies may be composed of other policies (i.e. nested), such that the condition of a policy includes the satisfaction of another policy. For example, 'Sensor(LightX) == true' may be a policy as well, such that, *"If and only if beamX_1 is interrupted, then Sensor(LightX) = true"*

```
Sensor(LightX) = true iff
    Interrupted(beamX_1) == true
```

In this manner, increasingly abstract and complex policies can be formed incrementally, making situation modeling through the use of policies more efficient from a programming perspective.

In the above examples of policies, the actions of setting a flag equal to true or executing a light activation function are explicitly stated actions. Policies can also be formed such that actions become implicit. For example, the action of the following policy is implicit. *"If and only if the sensor for lightX is true and the color 'color_Z' is set, then activate lightX to the color 'color_Z'"*

```
Activate_Lights(lightX, color_Z)iff
  Sensor(lightX) == true and
  Color_set(colorZ)
```

Since in this policy, the color that the light is activated to depends on what color has been set to the variable 'color_Z', the actions of the policy are not explicitly stated, but rather are implicitly stated.

Policies within the AmbiTalk system are context-based. To achieve context-based policies, the AmbiTalk system incorporates the use of *primitive context* [17] into the conditions of policies. Primitive context is a description of some distinct adaptation capability of a system. For example, *volume level* may be interpreted as a primitive context of a stereo system, such that the stereo system can be adapted to different levels of volume. A primitive context is called *active* when it describes the current state of a system. For example, the volume level of a stereo system can be in various states, whereby *high_volume*, *medium_volume*, and *low_volume* may be interpreted as active states of the primitive context volume level. Moreover, every primitive context is supported by a respective ontology, such that its ontology "describes the capabilities, relations and other information that are valid within that primitive context." [17] Therefore, an active primitive context may itself be considered a policy, such that every state of primitive context requires a set of conditions to be satisfied before becoming active. The interaction and negotiation of AmbiTalk agents is solely based on the exchange of active primitive contexts as *policy proposals.* The AmbiTalk system defines a policy proposal as a set of one or more active primitive contexts. When one AmbiTalk agent sends a policy proposal to another, it is proposing that agent to enter the particular system state represented by the active primitive contexts. Behavior reasoning arrives from an agent's attempt to include an active primitive context as one of its policies. For example, a primitive context utilized in the testing of the AmbiTalk system is *communication level*, which is defined to represent the level of communication discretion required by a location in terms of alert settings (e.g. ring, vibrate) and communication modes (e.g. chat, voice, video). Active communication levels range from *passive* (e.g. off), *semi-passive* (e.g. vibrate alert, chat conference), *semi-aggressive* (e.g. regular volume alert, voice conference), to *aggressive* (e.g. high volume alert, video conference). Following this representation, a public cafeteria might allow an aggressive active communication level, since communication discretion is not a major issue, while a meeting room might be restricted to a semi-passive active communication level, in order to control device-based disruptions and distractions. Therefore, moving from a public cafeteria to a meeting room is a change in active primitive context, and hence, a change in location policy. Primitive contexts can be used to represent the state of a system regarding many different aspects, including privacy, safety, security, and quality of service (QoS).

C. *Discovering location policies*

Within the AmbiTalk architecture the AT-UA must be able to discover the policies of its current location. It is possible to use a centralized approach, such that a centralized server covering a specific granularity is able to disseminate respective location policies. The advantage of this approach is that the AT-UA has the option of being aware of locations at varying granularities (i.e. room, building, block, city, etc.). The disadvantage of this approach is that the AT-UA must discover its physical location prior to requesting policies from the server. Since the AmbiTalk architecture only concerns itself with the location in the immediate proximity of the AT-UA, it takes a more direct approach and utilizes Bluetooth [6] as a direct method of interacting with the location.

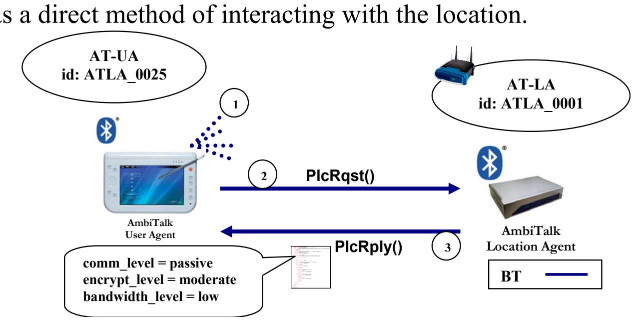

Fig. 3. BPS message flow

Figure 3 shows the message flow for discovering location policies using the developed Blue-Poll Scheme (BPS). The Bluetooth server's AT-LA is essentially a uniquely identifiable Bluetooth entity. The AT-UA continuously scans for Bluetooth entities. When the AT-UA enters a new location, it is able to discover the Bluetooth server's AT-LA (Action 1), since all AT-LA Bluetooth identifiers contain a common string of characters followed by a unique sequence of numbers (e.g. AT-LA_0000). An AT-LA identifier that is different from the one previously recorded by the AT-UA, implies that the identifier represents a new or modified location. When the location modifies any of its policies (i.e. active primitive contexts), the AT-LA must change its identifier. To support this method of discovering fresh AT-LAs, all locations in a calculated granularity have a unique pair of identifiers available to them. Once the AT-UA discovers a fresh location, it uses Bluetooth to query the AT-LA for the location's policies (Action 2). The AT-LA then replies with a policy proposal (Action 3).

D. *Behavior reasoning through policy proposals*

Within the AmbiTalk architecture, the AT-UA must know how to react to a policy proposal. In order for the AT-UA to make an intelligent behavioral decision, it must have its own context-aware system. Since a requirement of the architecture is to keep mobile devices light weight in terms of sensors and context gathering, a micro context-aware system (mCAS) function is developed. The mCAS function compliments the more complex CAS found in a location by exploiting the acquisition of active primitive contexts. The only environmental context required by the mCAS function is the Bluetooth discovery of active primitive contexts. All other context information is profiled or derived locally within the AT-UA.

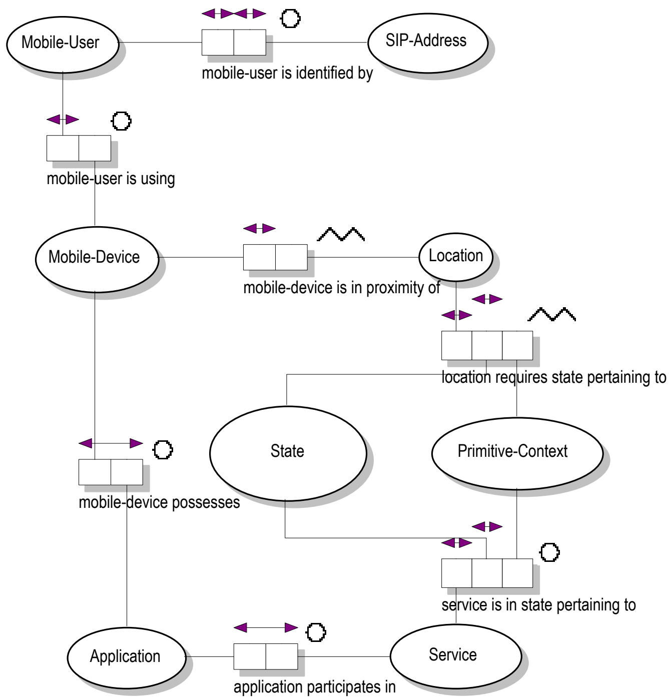

Fig. 4. General mCAS context model

Figure 4 shows the general context model used by the mCAS function. The context model is developed using the modeling technique proposed by McFadden et al [18]. The technique is an extension of Object Role Modeling (ORM) [19]. ORM views the world simply in terms of objects playing roles. Objects are represented as ovals participating in relationships through the joining of role boxes with arrows above them identifying relationship constraints. McFadden et al extend ORM to allow the representation of *sensed*, *profiled*, and *derived* information. This information is presented in the model as circle-shaped symbols (i.e. profiled context), m-shaped symbols (i.e. sensed context), and a star-shaped symbols (i.e. derived context). Profiled context is acquired locally from the device, sensed context is acquired from the environment and derived context is acquired from the aggregation of various profiled and/or sensed contexts.

Figure 4 depicts the relationships between eight objects: Mobile-Users, SIP-addresses, Mobile-Devices, Locations, Applications, Services, Primitive-Contexts, and States. The concept expressed within the context model is as follows. A *mobile-user* is a person uniquely identified by a *SIP-Address*, currently using a *mobile-device* in the proximity of a particular *location*. A mobile-device holds a variety of *applications*, which provide a range of *services*. The behaviors of a mobile-device, service, or application are direct responses to the current active *state* of a *primitive-context*, whereby a location requires the adherence of a particular active state of a primitive-context. Policies can be represented as derived context within the context model; however, they have been left out to give a clear general view of the context model.

E. A model for ubiquitous interaction

The AmbiTalk architecture enables bilateral interaction between the AT-UA and the AT-LA, as well as between the AT-UA and a second AT-UA. To satisfy these interactions, the Ubiquitous Interaction Model (UIM) is developed. The model is presented in Figure 5 as a UML state diagram.

AT-agents (AT-UA, AT-LA) are in a listen state until they acquire a policy proposal. All interaction is achieved through the exchange of policy proposals. A policy proposal is classified as outbound if the proposal is composed by the AT-agent itself and is being sent out to another AT-agent. A policy proposal is classified as inbound if it has been received from another AT-agent. The ability for AT-agents to both send and receive policy proposals allows for actions of negotiation.

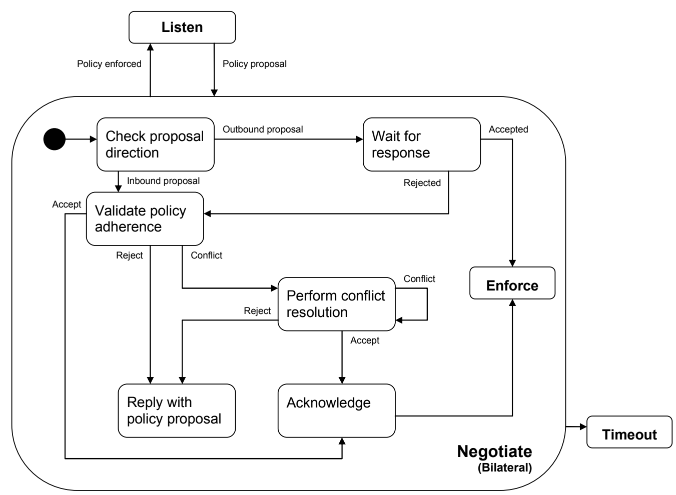

Fig. 5. UIM

Outbound proposal: When an AT-agent_1 makes a policy proposal, it waits for a response ("Wait for response"). If the proposal was successful and the proposed AT-agent_2 acknowledges its adherence, then AT-agent_1 returns to a listen state ("Enforce" then "Listen"). However, if the AT-agent_2 cannot adhere, then it has responded with its own proposal implying a rejection ("Validate policy adherence").

Inbound proposal: When an AT-agent_2 receives a policy proposal, it first validates its ability to adhere to the proposal ("Validate policy adherence"). This is achieved by determining if it contains a set of policies that imply either the ability/inability to comply, or a *policy conflict*. If a policy conflict has occurred, then it performs *conflict resolution* ("Perform conflict resolution"). There are many different approaches to conflict resolution in the literature [20], which are often application dependent. An AT-agent, therefore, keeps looping through different methods of conflict resolution until either the conflict has been resolved or a timeout occurs. To achieve conflict resolution, the AmbiTalk system incorporates policy context [21]. Policy context is a logical point within the interaction model such that additional information can be generated to resolve conflict. For example, upon a situation validating two policies, which are unable to coexist, policy context is able to give preference to one of the two actions. Policy context can be implemented within high level policies to resolve conflict.

Once AT-agent_2 can conclude its ability to comply or not, then upon a resolution to comply, it acknowledges its compliance ("Acknowledge") and proceeds to enforce the behavior decision ("Enforce"). Should it reach a non-comply resolution, then it responds with its own policy proposal ("Reply with policy proposal"); hence, a negotiation process begins. A negotiation process is included into the model to allow two AT-UAs to negotiate a session, which is discussed in Section VI-B. There is no negotiation between the AT-UA and the AT-LA, as in the AmbiTalk system the AmbiTalk device must adapt to the policies of the location.

VI. Communication Adaptation Architecture

There are two situations in which communication adaptation can take place and they are 1) pre-call: before a media session is established and 2) mid-call: during an active media session. Both of these situations require the negotiations of AmbiTalk user agents since each AmbiTalk device must respect the policies of its respective location. This section discusses pre-call adaptation, followed by the negotiation method used by the AmbiTalk user agents, and then mid-call adaptation.

A. Pre-call adaptation

The AT-UA's mCAS function contains local policies defined by the system and influenced by user preferences. It is specified in Section IV.A that the AmbiTalk system should be flexible in terms of various behaviors satisfying the policies of the location. This range of flexibility allows a user to influence and set preferences for certain behaviors. For example, when entering a meeting room, a user might prefer their device to switch to a dormant state (e.g. hibernate) as opposed to entering a discrete, but still operational, state (e.g. silent). Moreover, a user could specify contact lists relevant to specific situations, dictating which contacts are able to reach them.

The AmbiTalk system achieves the adaptation of communication services through the joint effort of the AT-UA's Service Functionality Controller (SFC) function and SIP User Agent (SIP UA) function. The SFC function manages the communication services available on the AmbiTalk device and the SIP UA function behaves as a SIP user agent. Since the SIP UA function is responsible for the initiation and maintenance of media sessions, it must be aware of what communication services to invoke. To avoid modifying the SIP UA function past the traditional functionality of a SIP user agent, the actions of communication adaptation are kept hidden, such that the SIP UA function turns to the SFC function for service activation. The SFC function initiates, modifies, and terminates services in an effort to only incorporate services into the media session that, through the exchange of policy, have been designated as part of the AmbiTalk device's current required behavior. For example, the SFC function could switch between various alert settings, communication modes, encryption levels, volume settings, wireless network types, and more. This makes the issue of pre-call adaptation fairly straightforward, as it simply involves the SFC function modifying the overall communication functionality. Therefore, when the policies of the location change or a new location is entered, using the Blue-Poll scheme the AT-LA's mCAS function discovers the policies of the location, reasons behavior based on the policy proposal, and instructs the SFC function on how to modify the communication functionality.

B. Call negotiation

The SIP UA function is designed to communicate all SIP requests and replies pertaining to session management to the AT-UA's Controller function, and wait for management instructions relating to when and how to initiate/maintain/terminate a SIP session. In this manner, the AT-UA's controller function is able to utilize SIP as a tool for call negotiation.

The interaction of AmbiTalk devices is the primary case of bilateral interaction. In this case, there are two AmbiTalk devices, each having modified their behavior to respect the policies of their current locations. The effect of this is that, when these two devices attempt to initiate a communication session, each will enforce the policies of their respective locations. A conflict may now arise if the policies of the two locations differ. To account for this, AT-UAs are enhanced to exchange policy proposals, permitting the two AT-UAs to negotiate the policies of a media session. This is achieved by two AT-UAs exchanging a set of active primitive context value pairs using the header of the traditional SIP INVITE request. The existing SIP INVITE header is modified to include a policy heading, whereby the active primitive context value pairs are placed.

```
INVITE sip:cn@ambitalk.com SIP/2.0
Policy: <comm_level,passive>, ...
To: <sip:cn@ambitalk.com>
From: <sip:mn@ambitalk.com>
```

As another option, active primitive context value pairs could be incorporated into the Session Description Protocol (SDP) [22] body within an INVITE request. SDP is a standardized method for describing session-relevant media capabilities, among other information.

When the AT-UA receives an INVITE request, UIM is triggered. Following the UIM flow (see Figure 5), the AT-UA's mCAS function validates a policy proposal by comparing each value of the proposed active primitive contexts against its own. Since the values of active primitive contexts are defined within the AmbiTalk system as specific levels (e.g. comm_level = passive), the mCAS function is able to map these levels to integer values (e.g. aggressive = 5, passive = 1) and perform arithmetic comparisons. As a result of these comparisons, the mCAS function can decide, in terms of each active primitive context, which communication party is more restricted. For example, if a device_1 is in a location requiring comm_level = 5 (i.e. aggressive), while a device_2 is in a location requiring comm_level = 1 (i.e. passive), then device_2 is the more restricted party. As a general system policy, the active primitive contexts applied to the media session should respect the more restricted party. Following the previous example, if device_1 sends a policy proposal to device_2, then device_2 will reject the policy proposal, since it cannot adhere to an active communication level of aggressive. However, if the policy proposal is reversed, such that device_1 receives a policy proposal from device_2, then device_1 will accept the policy proposal, since it can adhere to an active communication level of passive.

When the mCAS function decides to reject a policy proposal, the controller function instructs the SIP UA function to reject the INVITE request and send an INVITE request embedded with its own policy proposal. The new INVITE request is addressed using the information taken from the rejected INVITE request. The actions of rejecting and proposing policies (i.e. negotiation) lead both SFC functions to activate a media session that respects the location policies of both mobile nodes. This negotiation process occurs every time the policies (i.e. behavior) of either devices change, such that one or more of the policies adopted by the media session are no longer appropriate.

C. *Mid-call adaptation*

For the AmbiTalk device to change behavior while in an active call, the AT-UA's SIP UA function may be required to adapt the current media session, as the media session may no longer be appropriate. Therefore, the SIP UA function is enhanced by the development of a SIP Session Adaptation Mechanism (SSAM).

The adaptation of a media session requires the SIP UA function to terminate the current SIP session and initiate a new one, such that the SFC function has modified the services to be incorporated into the media session. It is possible to perform media adaptation without the termination of the current SIP session; however, due to session policy negotiation (discussed in Section VI.B), session termination is included into the mechanism. The SSAM flow for adapting an active media session is presented in Figure 6. The flow begins with the controller function instructing the SIP UA function to send a BYE request (Action 1), which terminates the current SIP/media session (Actions 2 and 4). After the SFC function terminates the media session, the controller function instructs it on how to modify the services (Actions 5 and 6). Once the controller function receives a 200 OK message (Action 8), it instructs the SIP UA function to initiate a new SIP session (Action 9), supplying the SIP UA function with the addressing information recorded from the terminated session. When the SIP UA function initiates a new SIP session, it signals the SFC function to activate the modified incorporated services (Actions 13 and 14), which starts a new media session.

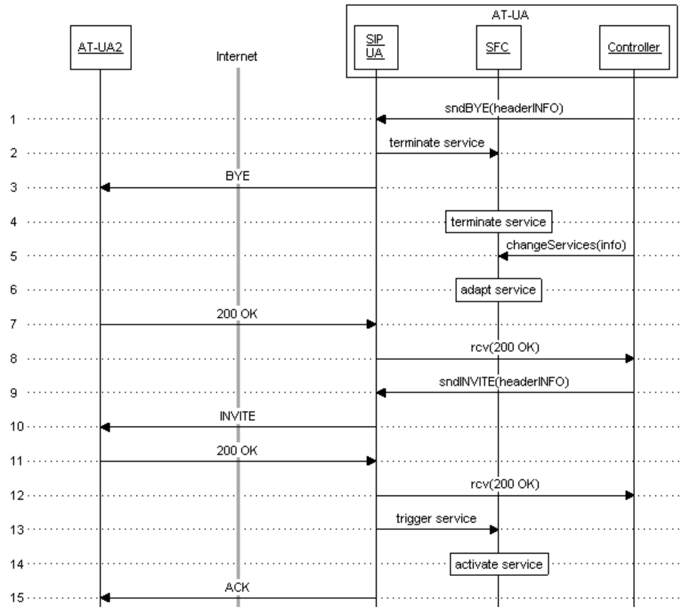

Fig. 6. SSAM

VII. Prototype

The complete AmbiTalk architecture is built upon the Microsoft Windows XP platform. AT-components (AT-UA, AT-LA, SFC, mCAS) are developed using the Java 2 Standard Development Kit (J2SE SDK) [23]. The Bluetooth application within the AT-UA and AT-LA components are developed using Blue Cove [24], an open source implementation of the JSR-82 Bluetooth API for Java [25], supporting the Windows XP SP2 Bluetooth stack [26]. The SIP user agent used is Columbia University's SIPc [27]. The communication services incorporated within the prototype are Videoconferencing Tool (VIC) [28], Robust Audio Tool (RAT) [29], and Whiteboard (WBD) [30]. Karmouch [31] presents the complete details of the prototype.

In order to provide a proof of concept of the AmbiTalk system, a series of test cases regarding communication discretion are derived and executed on the prototype system. Karmouch [31] presents the derived test cases and a discussion of their results. Summarizing, the results show the ability of AmbiTalk devices to ubiquitously adapt their behavior through the use of the BPS, UIM, and SSAM AmbiTalk solutions. Furthermore, the results show Bluetooth and SIP methods of ubiquitous communication as being successfully integrated into the architecture, such that the AmbiTalk devices are able to ubiquitously communicate with each other as well as with AmbiTalk locations. Moreover, the AmbiTalk devices show an implicit awareness of the user's environment.

VIII. Conclusions and Future Work

The system proposed in this paper is an attempt to push device-mediated communication into the paradigm of Ambient Intelligence. The AmbiTalk system incorporates context-awareness into the device-mediated communication experience. Moreover, the AmbiTalk system is capable of continuously adapting the behavior of a mobile device to respect the contextual environment of its user.

The presented AmbiTalk architecture was conceptually divided into the two design concepts of Policy Discovery/Reasoning and Communication Adaptation. Developed to satisfy the first concept was the Blue-Poll Scheme (BPS), the Ubiquitous Interaction Model (UIM), and the Micro Context-aware System (mCAS). Developed to satisfy the second concept was the SIP Session Adaptation Mechanism (SSAM). In order to show research value, the implementation details of the AmbiTalk prototype were provided and its test results were summarized. Moreover, the prototype showed the ability for locations and users to enforce policies related to communication discretion.

Areas of future work involve the further exploration of applicable conflict resolution techniques, the expansion of the prototype to include the Microsoft Windows Mobile and Palm OS platforms, and the implementation/testing of multiple primitive contexts relating to user privacy, security, safety, and quality of service (QoS).

References

[1] Ashraf Khalil and Kay Connelly, “Context-aware Configuration: A study on improving cell phone awareness,” in the proceedings of Context 05, Paris, France, July 2005.

[2] Mahesh S. Raisinghani, Ally Benoit, Jianchun Ding, Maria Gomez, Kanak Gupta, Victor Gusila, Daniel Power and Oliver Schmedding, “Ambient Intelligence: Changing Forms of Human-Computer Interaction and their Social Implications,” Journal of Digital Information, Volume 5 Issue 4, Article No. 271, 24 August 2004.

[3] Rosenberg J., Schulzrinne H., Camarilli G., Johnston A., Sparks R., Handley M., and Schooler E., “SIP: Session Initiation Protocol,” IETF RFC3261, 2002.

[4] 3GPP, “IP Multimedia Subsystems (IMS) (Stage 2) – Release 6,” TS 23.228, September 2002. [online] available: ftp://ftp.3gpp.org/Specs/archive/23_series/23.228/

[5] Henning Schulzrinne and Elin Wedlund, “Application-Layer Mobility Using SIP,” ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, Issue 3, pp.47-57, July 2000.

[6] J. Haartsen, M. Naghshineh, J. Inouye, O. Joeressen, and W. Allen, "Bluetooth: Vision, goals, and architecture," ACM Mobile Computing and Communications Review, vol. 2, pp. 38--45, Oct 1998.

[7] Mika Raento, Antti Oulasvirta, Renaud Petit, and Hannu Toivonen., “ContextPhone: A Prototyping Patform for Context-Aware Mobile Apllications,” in the proceedings of the IEEE CS and IEEE ComSoc, pg. 51-59, April-June 2005.

[8] Siewiorek, D., Smailagic A., Furukawa, J., Moraveji N., Reiger K., and Shaffer J., “ SenSay: A Context-Aware Mobile Phone,” in the proceedings of the 7th IEEE International Symposium on Wearable Computers, New York, USA, October 21-23, 2003.

[9] Kay Connelly and Ashraf Khalil, “Towards Automatic Device Configuration in Smart Environments,” in the proceedings of System Support for Ubiquitous Computing Workshopat the Fifth Annual Conference on Ubiquitous Computing, Seattle, Washington, 12 Oct. 2003.

[10] Manuel Gortz, Ralf Ackermann, and Ralf Steinmetz, “Enhanced SIP Communication Services by Context Sharing,” in the proceedings 30th Euromicro conference -component-based software engineering track, Rennes, France, September 1st -3rd, 2004.

[11] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., “RTP: A Transport Protocol for Real-Time Applications,” IETF RFC 3550, 2003.

[12] Hui Lei, Daby M. Sow, John S. Davis II, Guruduth Banavar, and Maria R. Ebling, “The Design and Applications of a Context Service,”ACM SIGMOBILE Mobile Computing and Communications Review, Volume 6, Issue 4, October 2002.

[13] Baldauf, M., Dustdar, S., and Rosenberg, F, “A Survey on Context-aware systems,” International Journal of Ad Hoc and Ubiquitous Computing, to be published.

[14] Thomas Strang and Claudia LinnhoffPopien, “A Context Modeling Survey,” in the proceedings of the First International Workshop on Advanced Context Modeling, Reasoning and Management, UbiComp 2004, Nottingham, England, 7-10 Sep. 2004.

[15] Joseph Y. Halpern and Vicky Weissman. "Using First-Order Logic to Reason about Policies,"in the proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW'03), p. 187, California, USA, June 29 - July 2, 2003.

[16] Ora Lassila, “Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help?,” in the proceedings of the Semantic Web and Policy Workshop, 4th International Semantic Web Conference, Galway, Ireland, 7 November 2005.

[17] Andreas Pappas, Stephen Hailes, and Raffaele Giaffreda, “A design model for context-aware services based on primitive contexts,” in the proceedings of the First International Workshop on Advanced Context Modeling, Reasoning and Management, Tokyo, Japan, 11-14 September 2005.

[18] Ted McFadden, Karen Henricksen, Jadwiga Indulska, and Peter Mascaro, “Applying a Disciplined Approach to the Development of a Context-Aware Communication Application,” in the proceedings of the 3rd Annual IEEE Int’l Conf. on Pervasive Computing and Communications, Hawaii, USA, March 8-12, 2005.

[19] T. A. Halpin, “Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design,” Morgan Kaufman, San Francisco, 2001.

[20] Evi Syukur, Seng Wai Loke, and Peter Stanski, “Methods for Policy Conflict Detection and Resolution in Pervasive Computing Environments,” in the proceedings of the Policy Management for the Web Workshop, 14th International World Wide Web Conference, Chiba, Japan, 10 May 2005.

[21] M. Smirnov, “Autonomic Communication: Research Agenda for a New Communication Paradigm,” Fraunhofer FOKUS, p. 9, November 2004.

[22] Handley, M., and Jacobson, V., “SDP: Session Description Protocol,” IETF RFC 2327, 1998.

[23] Java(TM) 2 SDK, Standard Edition [online] available: http://java.sun.com/j2se/

[24] Blue Cove project [online] available: http://sourceforge.net/projects/bluecove/

[25] JSR-82: Java APIs for Bluetooth [online] available: http://www.jcp.org/en/jsr/detail?id=82

[26] Windows XP Service Pack 2 [online] available: http://www.microsoft.com/windowsxp/sp2/default.mspx

[27] Xiaotao Wu and Henning Schulzrinne, "SIPc, a multi-function SIP user agent," in the proceedings of 7th IFIP/IEEE International Conference, Management of Multimedia Networks and Services (MMNS'04), pp. 269-281, ,San Diego, CA, USA, Oct. 2004.

[28] UCL Network and Multimedia Research Group, Videoconferencing Tool (VIC), [online] available: http://www-mice.cs.ucl.ac.uk/multimedia/software/vic/index.html

[29] UCL Network and Multimedia Research Group, Robust Audio Tool (RAT), [online] available: http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html

[30] UCL Network and Multimedia Research Group, Whiteboard (WBD) tool [online] available: http://www-mice.cs.ucl.ac.uk/multimedia/software/wbd/index.html

[31] Eric Karmouch, “Design and Implementation of AmbiTalk: A Ubiquitous System for the Automatic Adaptation of Mobile Communication,” M.Sc. Thesis, School of Computing, Queen’s University, April 2006.