Good example of context-aware middleware for mobile social networking research pa...

Business, Management

\n[toc title="Table of Contents"]\n

\n \t

\n[/toc]\n \n

Context-aware Middleware for Mobile Social Networks

Abstract

Introduction

Since the early days, people have been sharing information about their activities ranging from social activities, professional activities and individual lives in different ways. Schools, bars and restaurants have been the most favorable places for sharing this information before the development of information technology. Technology has changed completely how people communicate and share information. Technological advancements in wireless networks coupled with widespread use of portable devices have offered a unique opportunity for improving social networking services. The diverse use of Hoc and mobile networking solutions in many areas has led to a paradigm shift from fixed to mobile infrastructure. Currently people can share videos, discuss news, share professional reports and data and read other people’s opinions anytime anywhere.

A good social computing platform should be able to separate the concerns of managing the social network with the requirements of the application. The platform should also be able to integrate common management functions for personalization of social networks that are location dependent. The platform should also be able to propagate the visibility of social networks up to the application level. The different platforms that have been successfully implemented include CAMEO, SOMOA, ICAM, CARISMA and CAPNET.

Mobile Ad Hoc Networks (MANETs) are gaining a lot of popularity due to the development of wireless devices. MANETs is widely used due to the complexities associated with wired networks. MANET networks work hand in hand with other networking technologies to form interactive applications. MANETs is easier to deploy and have a low development costs hence, their widespread use. MANETs solve most of the network related issues. They don’t have a central management and control Centre hence users can easily set their preferences .

MANETs are very volatile due to their mobility, and they have a high dynamic topology. These variations in topology cause a series of network instabilities, which reduces the performance of applications on this network. The concept of context-awareness can be used to build reliable and predictable MANETs by providing information about the context of the mobile node which can be used to predict the mobility patterns. MNETS use this patterns to adjust in order to prepare for topological variations hence, maintain some degree of stability. Intelligent Content Aware Middleware (ICAM) can be deployed to offer functionality support.

The middleware platform in a MANET network is divided into two views; that is context related and middleware related. The context issues include capturing the content, processing the content and extracting knowledge from varied sources of contextual information. The middleware provides a platform for communication for different users within the network. The dynamism and network instability are given support by the middleware. Gathering of contextual information is done by various sensors that are being controlled by mobile devices. The middleware should be flexible so as to accommodate all types of sensors including the ones that are currently not supported. Mobile devices are designed to have limited storage, energy and the processing power. This issues are usually addressed by the middleware, hence a proper and dynamic middleware should be developed. Representation of the context is of great importance hence the model developed should be able to enable exchange of information and uniquely manage different types of contexts. Ontologies should be used to enable interoperability among contextual domains.

The system should have a given level of programming so as to increase the efficiency of resource use and communication. The middleware should be able to handle data and task distribution roles where the mobile node has insufficient resources to handle such tasks. The middleware should also be light so as to load on mobile devices with minimum resources. A well designed middleware should be reliable and robust and support parallel user transparency and interoperability.

ICAM lies between the networking and application layer of the network. Due to its advanced functionality, ICAM is able to configure these layers. Context information is considered to make necessary configuration changes. Network communication is not influenced by ICAM and among the design considerations is that ICAM should be transparent enough. ICAM is usually idle in the background of the mobile node and only becomes active when contextual information collected indicate that there is a need to modify either the application or networking layer. ICAM continuously monitors all the sensors available to the mobile device and manages the communication interface between these sensors and the mobile device.

ICAM is developed using java’s J2ME because of the design considerations. Contextual information is stored using XML simply because XML describes the different contextual sources using a single XML schema. XML offers great interoperability and high performance. XML is also the best suited in attaching semantic information to contextual data.

Another middleware solution is the Socially Aware and Mobile Architecture (SAMOA) which has a set of common management tools for personalizing social networks that are location dependent. SAMOA also propagates visibility of a social network up to the application layer. The framework supports the development of anytime, anywhere content aware social networks that group users of a given locality together and share other interests such as attitudes, social interests and likes. SAMOA empowers mobile users to create roaming profiles for their social networks. The middleware then follows the activities of that particular user recording each instance of interest.

SAMOA basically focuses on the user that is the visibility of the physical location of the user and features of that location. Place visibility regulates the scope over which extraction of social network for the entities that are in the same location as the user. The visibility of the user is used to define the range of discovery and then this data is used to create personalized social networks. SAMOA also does modelling of data and then represents it in the form of semantic metadata. The data is then analyzed using semantic matching algorithms and then checks for possible semantic compatibility. The social network management model developed by SAMOA differentiates between three roles of management. The first management role is the mobile users who have an interest in creating social networks. They define their own profiles that will be used in tracking their activities. They also define the boundaries that guide the extraction criteria. The second role is the clients, who are users within the locality of the manager and qualify to become part of the social network of the manager. The last role is that of other members that belong to the specific social network. A mobile user can play any role at a given instance. A manager can exploit two social networks that is; the place dependent social network that only allows the manager to view members currently within his/her locality and the global social network which records all the records as the manager moves from one location to another. The model uses two semantic matching algorithms to determine location dependent social networks. The SAMOA social networking management model basically focuses on a place. This enables us to develop well defined boundaries for discovering patterns of users.

The SAMOA middleware is a two-layered architecture that is built on top of the java virtual machine. It has the social network management layer and the basic service layer. The support for communication, naming and the detection of entities in the same locality is facilitated by the basic service layer. Implementation of point to point and multipoint communication patterns based on UDP is done by the message transport manager (MTM).

The MTM provides the path for communication between different entities. This communication follows a broadcast protocol that is designed for cell-based environments and the flooding protocol used for MANET settings. Generation of personal and place identities is done by proximity manager. This manager adopts a naming approach which ensures statistically that the identifiers are unique. The location managers also allows entities to broadcast their online availability when the user is online through advertisement messages. These messages comprise of the entity’s place and personal IDs. SAMOA then uses these messages to build a table of online users in relation to their locality.

The profile manager contains the necessary tools for checking for profile correctness and then supplies these profiles to entities that match. The ontology editor in java compiler provides a platform for profile browsing and visualization. Several overheads arise from the use of middleware to support extraction of social network, which depend on performance of middleware facilities and the environment where they are deployed.

The main issues reported with the use of SAMOA is privacy. The developers are trying to integrate the algorithm with security support algorithms to address the issue of privacy. Unless they address this issue, it may fail to pick up after implementation.

Another middleware is CAMEO (Context Aware MiddlEware for Opportunistic mobile social networks). CAMEO basically provides functionality for management of context, optimized set of services and protocols for networks that are opportunistic such as content-aware forwarding protocols. CAMEO also supports the development of context aware application in an optimized manner. CAMEO is contained in two packages; that is the Local Resource Management Framework (LRM-Fw) and the content Aware Framework (CA-Fw). LRM-Fw is concerned with implementation of features that interact directly with local resources. CA-Fw is responsible for the storage, analysis and distribution of all the context information. CA-Fw also defines optimal forwarding protocols.

CAMEO starts by collecting and managing of the social and local context of each mobile node. CA-Fw defines procedures for distribution of information among 1-hop neighbors. The procedures are optimized based on identification of context components by employing the harsh tag to avoid periodic distribution of the information. CAMEO receives periodic updates of the local node that contains the user profiles and application context and those of the user’s neighbor.

CAMEO uses context based utility function to optimize information requested by an internal service. For the case of context aware forwarding protocol, the context manager is first tasked to analyze the context information of the neighbors currently. It then evaluates the probability of each mobile node to deliver the message to the final destination.

CAMEO defines an application protocol interface for interactions with each applications. An inter process communication (IPC) channel is established with each application so as to collect context information of that application and send the message when it is required. CAMEO also defines a callback interface for notification of occurrence of an event due to initiation of a given protocol. The callback interface is defined using android interface definition language (AIDL).

The user profile module interacts with CAMEO that require the user to provide initial information about his/her profile and authorize distribution of information over a social network. CAMEO also works with android systems so as to collect information and then establish a communication channel over the channel. Mostly mobile social network applications that are developed using CAMEO have been successfully implemented on android devices only. The communication between applications and the middleware is achieved using android java proxy class that implements the transfer of parcelable objects between IPC paradigm and the processes.

Another middleware that can be employed for context aware systems is CAPNET. CAPNET offers greater mobility of software components. It is placed on top of the existing technologies but below the application layer. Different functionalities offered by CAPNET can be subdivided into a set of components with each providing a specific service. These components include user interface, service discovery, connectivity management and context based storage.

All the specifications outlined by context awareness are fulfilled by CAPNET. It also offers support for devices that collect location data and record streamed videos. This data is distributed over a number of software components which are at a different location but communicate through a network. Abstraction of content is transparently performed by intermediate components hence the applications only receive processed data from these intermediate components. CAPNET also provides storage space for context data and files. CAPNET is also a very light framework that can load easily on mobile devices with limited bandwidth. Bandwidth and traffic of the network is controlled by the connectivity manager. Definition of network policies and procedures for traffic is managed by the application itself.

CAPNET middleware components share a common interface, hence hardware support can also be extended. These components are very small with well-defined interfaces. The implementation of these components is hidden behind the interface of the components. This enhances communication between the components. The components can also be started while the other applications are running. Messaging, management and discovery components is also contained in CAPNET. The components are assigned reference addresses after initialization. When a request is made to a specific component, a reference to the handle is passed instead of passing the actual object. When the device does not have enough resources to execute a component, the component manager decides to send the component to the remote host. This process is required more so when the media components want to load or execute multimedia content.

All communication within the CAPNET middleware is handled by the messaging component. This messaging component defines channels with asynchronous communication capability. Connectivity manager handles switching between channels according to specified internal policies. These channel policies can only be altered by the user or the application itself in the public interface. Connectivity management comprises of the policy control, connection control and connection monitoring.

MobiClique is a middleware platform that allows users to extend their online social networks while maintaining the current ones. MobiClique is different from the other solutions since it is independent of an infrastructure connectivity. The platform relies on connectivity between close proximity devices. MobiClique uses existing virtual user profiles to enable temporary connections between two users on a given location and social compatibility. Users have a choice to decide setting up new friendships or even share contexts.

MobiClique is a dynamic platform that explores beyond device to device communication. MobiClique offers a generic framework for implementation of different applications. Each user can rise in the ranks upon discovery of new contacts and other users. This platform provides collaborative forwarding based on the opportunistic contacts. MobiClique utilizes facebook API to access social graph data. This API contains different functions that perform different tasks such as retrieving of user profiles, friends list and activities the user is currently giving attention.

MobiClique is not yet fully developed, and it has limited applications as compared with the other platforms. The platform does not support adding of new friends and groups to the profiles hence there is the need to partner with third party software applications to help in the collection and updating of information to the network.

The architecture of MobiClique is in such a way that it operates in an ad hoc network. This implies that the devices are within the proximity though they can access the internet for synchronization of information between an open system network (OSN) and the device. Each device in the MobiClique platform is represented by a node. An existing OSN service is used to initialize and synchronize the node.

MobiClique nodes execute in a periodic loop and the process can be subdivided into three: discovering the neighborhood, identifying the user and exchanging data. The neighborhood discovery is achieved by using radio technology methods. Once the neighbors have been identified, the device opens a communication link that allows exchange of information between the two devices. This involves exchange of full profile information of users. Once the identification phase is over, exchange of application level messages takes place.

Coalition is a context aware middleware that was designed to offer homecare services and monitoring to the elderly and the sick at homes. Originally the middleware was designed to offer reusable, open and programmable infrastructure for social networking and data sharing. Coalition is considered a service oriented architecture. This implies that all functionalities such as acquisition of context data, analysis of context data and service discovery are deployed as services for the end users.

The architecture of Coalition consists of four layers. The physical state layer represents the sources of context data. This sources include computer devices, sensors and actuators. The physical layer manages the interaction of this sources through the physical state gateway (PSG). The context data management layer (CDML) is in charge of managing the physical space. Physical space is the physical quantities that are around an entity. CDML also helps in context data lookup. an SQL based application interface is placed on top of this layer to help the applications in context data acquisition. The service management layer carries out services discovery of both the system services and other third party services. The application layer manages the limits of the geographical area for discovery purposes. Content aware applications reside on top of the service management layer.

Context data management is achieved through the physical space model. This model provides a lot of flexibility to applications for access of data from sources. The physical space can derive data directly by sensors or logically analyze the physical space of the entity. Coalition offers mobility support by employing mobile space model. Mobile space is used for modelling and helps in data management of mobile entities.

CARISMA (Content-Aware Reflective mIddleware System for Mobile Applications) middleware was built to enhance content-aware, anytime mobile applications. These applications should be highly adaptive to enhance the performance of the model. Management of content changes is handled using built-in policies. The policies sometimes can have a common interest and lead to a conflict. Conflicts are handled using the micro-economic approach. This approach is based on sealed-bid auction. CARISMA does not base its development principle on transparency but on reflection. Transparency implies that the middleware are the ones involved in making decisions on behalf of the applications. Reflective systems allow the systems to modify themselves by means of adaption or behavior. This consideration is not so much considered in the other middleware discussed before.

Conclusion

There has been increased popularity of mobile devices in the due to advancements in technology. This has led to rapid investments of developing new applications. Several challenges have risen due to the quest to integrate the mobility of entities. Content-aware anytime mobile middleware platforms should be able to take care of deployment and run time configurations.

The content aware, an anytime mobile for social networks, has not been fully exploited and so far the existing solutions have some drawbacks since all of them take a different approach. More research should be done so as to integrate this different platforms so as to take the advantage presented by each while covering the disadvantages. Popularity of middleware platforms is rising since it solves many problems associated with user profiling and location identity. These platforms also try to separate different layers of communication to avoid conflicts and increase communication speed. A platform for validating content-aware, anytime for mobile social networks should be developed. This will assist developers to construct platforms that offer full functionality of content-aware applications.

References