Fusing Mobile, Sensor, and Social Data To Fully Enable Context-Aware Computing

Aaron Beach[1], Mike Gartrell[1], Xinyu Xing[1], Richard Han[1], Qin Lv[1], Shivakant Mishra[1], Karim Seada[2]
[1]University of Colorado at Boulder, [2]Nokia Research Center Palo Alto
{aaron.beach, mike.gartrell, xinyu.xing}@colorado.edu

ABSTRACT

In this paper, we identify mobile social networks as an important new direction of research in mobile computing, and show how an expanded definition of mobile social networks that includes sensor networks can enable exciting new context-aware applications, such as context-aware video screens, music jukeboxes, and mobile health applications. We offer SocialFusion as a system capable of systematically integrating such diverse mobile, social, and sensing input streams and effectuating the appropriate context-aware output action. We explain some of the major challenges that SocialFusion must overcome. We describe some preliminary results that we have obtained in implementing the SocialFusion vision.

1. INTRODUCTION

This paper is primarily a conceptual work that proposes how emerging mobile social networking technology can be systematically leveraged to enable a new and exciting generation of applications and computing that is more fully context-aware. We first observe that mobile social networks are likely to become a key future research direction for mobile computing. Mobile social networking is driven by the explosive growth of social networks like Facebook and Twitter, and the mushrooming popularity of mobile smartphones like the iPhone. To this, we add the rapid evolution of sensor networks, and broadly define a mobile social network as any distributed system that combines these three major components - mobile, social, and sensing - to create a contextual picture surrounding a user or group of users and offers services based on this context. Together, mobile, social, and sensing components provide the necessary location-aware, relationship-aware, preference-aware and sensory context to achieve Mark Weiser's vision of a comprehensively context-aware world of ubiquitous computing [1].

Mobile devices are nearly ubiquitous and can be leveraged to provide location and announce a user's identity/presence in a room or place [2, 3]. In addition, mobile smartphones have been enhanced with a variety of sensors, such as accelerometers, microphones, cameras, and even digital compasses, that can be mined to infer actions [4] or even orientation. However, mobile devices alone cannot provide a full picture of context, and in particular have a difficult time inferring an individual's tastes. On the other hand, social networks like Facebook, Twitter, MySpace, LinkedIn, and Orkut are rich with detailed contextual information describing an individual's personal interests and preferences as well as friendship relationships. However, social networks do not by themselves know the location of a user. Sensor networks provide a third dimension of input data that can further enrich the understanding of context, e.g. static sensor networks embedded in the local smart space can obtain temperature, humidity, infrared, audio, and video to more fully characterize an individual's context. Moreover, all three classes of mobile, social, and sensor data when temporally archived provide historical perspectives that even further enhance the understanding of context. It is the *combination* of these three key data streams that permits effective location-aware *and* preference-aware adaptation by software to effectuate a context-aware action in the vicinity of the user.

We believe that the scope of context-aware applications that will be enabled by mobile social networks spans the entire range of current applications, e.g. desktop, online, and mobile applications that will benefit by being enriched with context. For example, we have enhanced a traditional music player application with context-awareness to create a music jukebox that plays audio that is tailored to the tastes of an individual who is standing near the jukebox [2]. The mobile social network provides the location of the individual (near the jukebox) as well as the musical preferences of the individual (obtained from the social networking profile, which lists favorite bands, songs, etc.). Beyond this, we envision that mobile social networks will enable entirely new applications. For example, mobile social networking applications would give epidemiologists a better understanding of contact rates among individuals and how diseases propagate through a population, provided epidemiologists could access location data plus friendship and sensory information to strengthen potential contact inferences. Mobile social networks can also enable context-aware video screens, TVs and audio players. The film Minority Report illustrated context-aware video screens that morphed in real-time to present context-aware content that is tailored to the tastes of an individual walking by. This concept could be extended to groups, so that a family gathered around a TV at home is automatically shown programming that is suited to the family's preferences. We have prototyped a context-aware video player

that plays a movie based on the collective film tastes of individuals watching the same screen [5]. We contend that mobile social networks will make these applications, and much more, feasible and/or more effective by integrating holistic context in the form of location, preferences, relationships, and sensing.

We propose SocialFusion as a context-aware system capable of integrating these diverse data streams to effectuate context-aware actions both for individuals and for groups of people. The first contribution of the system is in its novel goal of integrating or fusing together such diverse mobile, sensor, and social input data. Prior research has addressed portions of the SocialFusion vision, but as far as we are aware, no single project has addressed the full scope proposed here for SocialFusion. Serendipity links together a stand-alone social network with mobile devices [3], but is not integrated with existing social networks like Facebook. WhozThat integrates Facebook with mobile phones to provide context-aware audio, but does not integrate with any sensors [2]. CenceMe mines mobile data provided by iPhone sensors to infer user actions, but is not integrated with social networks [6]. Early work on context-aware computing occurred largely prior to the advent of social networks [7, 8, 9]. A second contribution of SocialFusion is outlining an architecture to cleanly and systematically achieve inference and context-aware actuation. A third contribution is explaining some of the key challenges that SocialFusion faces. These are explained in the following section.

2. THE SOCIALFUSION SYSTEM

A key contribution of SocialFusion is that we describe how to systematically achieve the fusion of mobile, social, and sensing streams to generate context-aware output, not just for individuals but also notably for groups. We outline in Figure 1 how SocialFusion goes about this process of generating context-awareness by fusing mobile, sensor, and social data. The system architecture is organized into multiple stages and classes of data. We have organized the input data streams into three major classes, namely mobile data from smartphones, including location, accelerometer data, etc., sensor data from fixed sensor networks, and social networking data from Web sites like Facebook, Twitter, LinkedIn, MySpace, etc. This data is collected by the SocialFusion system, organized, processed, and filtered. In the second stage, a bank of classifiers operates on the assembled raw data to attempt to extract meaningful contextual inferences from the data, a process which we call "fusion". For example, one classifier may seek to determine the favorite musical style preferred by an individual by analyzing the personal patterns of behavior, while another classifier may seek to measure a characteristic of a group individuals, e.g. how closely related they are to one another in friendship. Based on these higher-level inferences as well as the raw data, in the third stage a recommendation engine employs data mining to generate context-aware output action(s). These are generated either for an individual or for a group of individuals depending on the circumstance.

A third contribution of this paper is that we identify the expected challenges faced by SocialFusion, why they are hard, and why existing approaches are inadequate to the challenge. We expect that some of the major challenges will include collecting and managing diverse data streams, making inferences to extract context for individuals and groups,
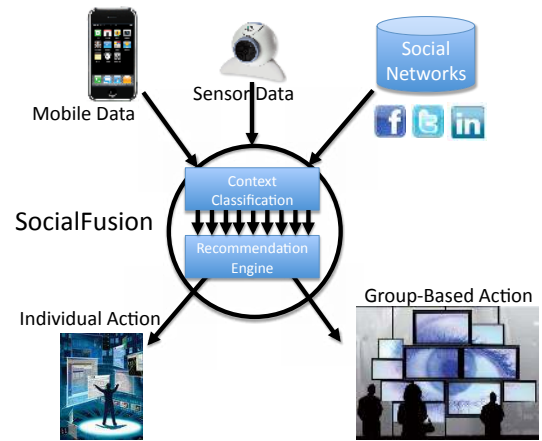

Figure 1: How SocialFusion fuses mobile, sensor, and social data to generate context awareness.

mining highly diverse data streams to generate individual and group-based recommendations, and preserving privacy and anonymity. These are described next.

2.1 Collecting Mobile, Sensor, and Social Data and Extracting Context

The first major challenge in building the proposed SocialFusion system is collecting and managing the input data. As mentioned, the system collects mobile data from users' mobile devices, sensor data from static sensor networks and social networking data from online sites. There are a number of difficulties that needs to be overcome to collect this data in an efficient manner. To collect data from users' mobile devices, a mobile application needs to be built that can identify the user's location and the phone's sensor values and pass on that information to SocialFusion. Given that a mobile device is typically used for a multitude of applications, this mobile application must be power efficient and power aware, preferably adaptive to the current remaining power of the device as well as non-intrusive to the user. Data from fixed sensors will need to be forwarded to SocialFusion either at regular intervals or whenever there is a major change in sensor values. This will require addressing the diverse APIs and data representations that different sensor networks support. Efforts to collect social networking data from online sites may be elaborate in some cases and require permission from the users, e.g. Facebook. In others, the data may be openly available, requiring only the building of components to import the data, e.g. Twitter. The resulting raw data may be noisy, and may need to be cleaned prior to hand-off to the next stage of processing. While earlier systems have utilized mobile, sensor or social networking data, it is the collection of data from a large number of diverse input streams that adds another level of complexity. For example, one concern about collecting information about a user from multiple Internet sources is that users typically use different usernames and email addresses for different Internet accounts. To address this, an identity aggregator will be incorporated, and location and preference information of each user is stored in a database organized by each user's aggregate identity. Since location information is time depen-

dent, it is important to have fresh estimates of user location, which are obtained primarily from the mobile component of SocialFusion, so the tradeoff between location update frequency and energy expenditure on mobile phones will have to be explored. Information about user preferences is relatively static, and we will update this information at regular intervals by accessing various relevant websites.

As the data from diverse input streams is collected, it needs to be appropriately organized and managed to enable efficient processing, e.g. to define and extract context. Managing data in the SocialFusion system is complicated because of three reasons. First, there are many different types of data ranging from different types of sensor data, varying location data, user preferences, and relationships. Second, since the data is collected in real time, it needs to be managed in real time. Finally, the amount of data to be managed is potentially very large, e.g. on the order of TBs. To support location queries or pattern matching, certain data will need to be efficiently stored using a high-performance relational database, while data which is simply queried should be optimized for scalability using a distributed database like Amazon SimpleDB. Appropriately classifying data this way and managing it efficiently is a crucial challenge.

The second major challenge is how to define and extract context from the multitude of data streams in the system. We are interested in understanding an individual's location, interests, tastes, orientation, and even emotional mood, mental state, and/or physical condition. Further, we are interested in group-related forms of all of these parameters. To the best of our knowledge, none of the earlier systems have attempted to define and extract context from such a multitude of data streams. We plan to incorporate a stage of context classifiers for each parameter of interest, each of which operate on the raw assembled data to produce a piece of context. Examples include a location classifier, a physical condition classifier, and group-based classifiers. A location classifier would combine information from both GPS positioning from a mobile phone and fixed sensor networking data to better pinpoint the location of an individual. A physical condition classifier would operate on both the accelerometer data provided by the mobile phone and the camera data provided by local fixed sensors to better evaluate whether an individual is in distress. The bank of classifiers is cleanly partitioned from the data collection mechanisms, and occurs as a stage subsequent to data collection.

Group-based classifiers operate on the raw data to identify group-based characteristics of interest. For example, suppose that the cell phones of John, Mike and Lisa's report that they are in close proximity to one another. This data by itself does not necessarily identify a group, because there may be others nearby. Suppose also that Facebook reveals that John, Mike and Lisa are good friends, i.e. we have a friendship classifier that operates on the social graph and classifies this group of individuals as good friends. It is the combination of these two sets of information fused together that gives us a much stronger indication that John, Mike and Lisa are spending time together as a group of friends.

We believe that the clean partitioning between data collection and data classification in the architecture of Social-Fusion will spur substantial new research in each area. For example, new research will need to be performed to develop accurate classifiers of emotional mood, physical condition, group activity, etc. In the case of the group identifica-

tion classifier, we have considered that phone accelerometers have been used along with activity recognition to identify groups [10]. Also, trajectory databases have been used to identify "convoys", i.e. groups of objects moving together in time [11]. Fusing together social network information (family/friend connections), simple activity recognition (such as sitting, standing, walking) and location information enables events such as church services or bowling leagues to be extracted as identifiable patterns which could be modeled over time as trends (or anomalies) in a model of group interaction. We are exploring a classification approach that groups users using fuzzy-clustering based on recent trajectories (location histories). These clusters would then be made distinct with social network and user-specific activity information. Users' history of group association would then be used to form probabilistic models, based on Markov chains, to represent their habitual social interactions (i.e. their temporal social structure).

2.2 Mining Diverse Input Data For Individual/Group Recommendations

As various types of mobile, social, and sensor data are continuously collected and inserted into the SocialFusion system, it is important that the system mines the data efficiently and effectively to discover inherent patterns and use them to generate appropriate output actions, such as playing a song that the user likes, or recommending movies to a group of friends in the same room. Our key challenge (and contribution) is to identify inherent patterns that span across large amounts of dynamic and diverse mobile, social, and sensor data, and combine such patterns for the most appropriate context-aware actions for individuals and groups.

To achieve this goal, we propose to first identify *frequent patterns* in users' mobile, social, and sensor data, including both *frequent itemsets* (set of items that co-occur frequently) and *frequent sequences* (items that often occur in a certain order). For example, through frequent pattern analysis of SocialFusion data, we may discover that John usually reads health-related news articles in the morning when he rides a bus to work; or that John, Mike, and Lisa usually go to a football game on Saturday and have dinner together afterwards. Our design improves existing frequent pattern analysis techniques by addressing a few unique issues. Due to the diversity in users' interests and daily activities, we first apply frequent pattern analysis to individual users and individual data sources of each user. This approach helps us to quickly prune out noisy and infrequent items in each category and focus on identifying and combining patterns that are truly frequent and span multiple data sources. In addition, the frequent pattern analysis process is incremental – as new data are added into the system, they are analyzed and integrated into previously-discovered patterns or are used to generate new patterns.

At run time, the frequent patterns identified above need to be combined with up-to-date mobile, social, and sensor data to identify the current context, whether an output action is needed/possible, and if yes, what types of recommendation to generate (individual vs. group, music vs. movie, etc.). Using the example above, when the system determines that John is riding a bus to work, the system will then try to recommend recent health-related news articles to John. Since the system has already identified the frequent patterns of interest, only certain contexts need to be detected and rec-

ommendations made, enabling very efficient output actions. A key design component in SocialFusion is efficient pattern management and pattern lookup such that unrelated data (and contexts) can be quickly filtered out.

One important contribution of SocialFusion is the capability of group-based, context-aware recommendation. Although different recommendation systems exist for individual users [12], groups of users [13], and specific data types, SocialFusion is the first system that enables group recommendation based on comprehensive mobile, social, and sensor data. Instead of a simple approach that aggregates multiple users into a single "virtual" user, SocialFusion supports a number of group-based satisfaction metrics, such as majority vote, maximizing minimal satisfaction, weighted influence of individual users in a group, weighted pair-wise influence, etc. The challenge is identifying different group characteristics and their impact on group decisions. For example, when the system detects a group of friends are likely to watch a movie together, to recommend movies to the group, the system considers not only the movie preferences of the individual users in the group, but also the group influence, which may be affected by leaders/experts in the group, pair-wise user similarity and/or social relationships. In the SocialFusion system, a number of independent recommendation techniques (e.g., Apache Mahout Taste) will be used to generate recommendations for individual users under different contexts; the system then combines the individual recommendations according to a specific group satisfaction metric. The system also will incorporate a feedback mechanism such that individual/group feedback regarding previous recommendations are utilized to improve future recommendations.

2.3 Preserving Security and Privacy

SocialFusion opens new avenues of research on how to allow a user to specify privacy while spanning diverse data streams. For example, a user may be willing to reveal their social networking preferences and location while shopping at the mall on the weekends, but not elsewhere. Users will need controls that easily allow them to balance their privacy needs with the utility of the application, namely which of these diverse information sources is revealed to whom and under what conditions. We intend to explore these issues as we move forward with SocialFusion.

Despite safeguards protecting both location and data privacy, a user's anonymity may still be compromised by revealing context-aware recommendations in public settings. Even though a participating user may have taken precautions in each of his individual data streams to independently shield certain information from others, e.g. financial or relationship status, the correlative power of SocialFusion in integrating multiple information sources may reveal in its recommendation potentially damaging or embarrassing information. Thus SocialFusion stimulates new research in designing algorithms that hide or anonymize such sensitive information so that context-aware output actions don't reveal more than is necessary. Our research is made even more challenging because we're dealing with diverse data streams.

The traditional approach to K-anonymity is to provide a guarantee over a data set such that quasi-identifiers associated with one identity in the data set are indistinguishable from (copies of) at least $k-1$ other identities in the data set. While interesting as related to data sets for research purposes, the traditional K-anonymity definition is not so useful when applied to the release of social, mobile, or other data through publicly available social networking APIs for the following reasons:

1. We cannot assume that all data sources (social, mobile, medical, sensor, etc.) will be fully anonymized. For example, we cannot assume that Facebook is going to guarantee anonymity over their entire data set and they are the only entity that could make such a guarantee.

2. It is possible that any or all social network attributes may be used as quasi-identifiers, therefore all profile data must be considered a quasi-identifier that could possibly be used for re-identification.

3. Public API calls (e.g., Facebook API) access or refer to only a small subset of the overall data set and usually only refer to a particular subset of attributes related to an individual. Whereas, traditional K-anonymity guarantees require anonymity across the entire data set.

Therefore we propose a related, but different K-anonymity problem defined as follows: *Given a partial release of data from a private data set, wherein all data is quasi-identifiable, the released data must map to at least k distinct sets of individuals within the data set.*

We have made progress on developing a new approach to K-anonymize diverse streams to preserve individual privacy. Prior work in K-anonymity seems unsuitable for context-aware mobile social networks because it assumes access to the entire data set, makes wrong assumptions about quasi-identifiers, or assumes that data may be distorted or generalized and still useful. For example, the algorithms may distort the data in some manner, either by introducing a random perturbation or transformation into the social graph [14, 15, 16], or by generalizing or "fuzzifying" the information prior to public release [17]. Such generalization may distort context-aware recommendations and output actions. We seek to develop a new class of K-anonymity algorithms that *selectively withhold* data, thereby preserving both the K-anonymity of the released data and its accuracy. We are thus developing K-anonymity algorithms that meet our new K-anonymity definition using optimized or "selective" withholding.

Our selective withholding approach to K-anonymity analyzes whether the release of a certain piece of data, e.g. a location or a social networking preference, will provide anonymity up to a degree of K. Consider the example shown in Figure 2. If the data set we wish to release is *(Chemistry class, Chris, 1)*, then the only possibility is *(Joe)*. However, if we withhold the data item *(Chris)* and only release *(Chemistry class, 1)*, then the possible identities are *(Joe)* OR *(Bill)*. Thus, we've increased the anonymity to $K \leq 2$. This example shows how we could selectively withhold certain data items to increase K to a desired value.

For more complex data sets, the Boolean expression linking the set of users to the set of data to be released needs to be systematically derived, which can be achieved using a directed graph that models the relationship between the data to be released and the users who could be linked to that
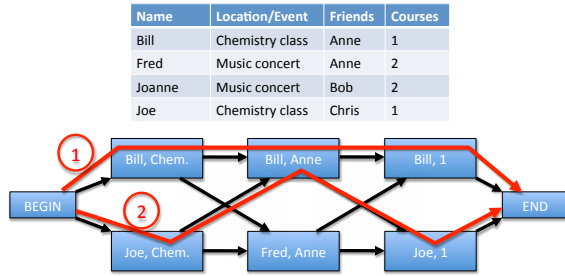
Name	Location/Event	Friends	Courses
Bill	Chemistry class	Anne	1
Fred	Music concert	Anne	2
Joanne	Music concert	Bob	2
Joe	Chemistry class	Chris	1

Figure 2: Our selective withholding approach to K-anonymizing data released from mobile social sensor networks. Logic minimization is applied to the Boolean expression generated from the data modeled as a directed graph

data. For any data set that we wish to release $(d_1, d_2, ..., d_n)$, we construct a column of nodes for each d_i, where each node in column i consists of the pair (username, d_i). This identifies all possible users who could be associated with the release of data item d_i. Next, we interconnect the nodes in a column with the nodes in the next column. This creates the directed graph. The set of all truth cases would be the superset of all paths from beginning nodes to the end nodes.

For example, before releasing *(Chemistry class, Anne, 1)*, we generate the directed graph shown in Figure 2 consisting of three columns, one each for all users associated with the Chemistry class, Anne, and one course. A single path #1 through the graph corresponds to the Boolean expression (Bill AND Bill AND Bill), or just (Bill). A second zigzag path #2 corresponds to the Boolean expression (Joe AND Bill AND Joe), or just (Joe, Bill). The union of all possible paths through the graph gives us all possible combinations of users that could be associated with the release of the data *(Chemistry class, Anne, 1)*.

Since the full Boolean expression could be quite complex, we need to devise an algorithm to quickly reduce the expression. We observe that logic minimization algorithms can accomplish this task. Several well known logic minimization algorithms exist, including ESPRESSO [18] and Quine-McCluskey. When applied to our example graph in the figure, the simplified Boolean expression reduces to (Bill) OR (Joe, Fred), thus illustrating the power of this technique. We determine if the simplified Boolean expression meets some K-anonymity guarantee by counting the number of terms in the simplified expression. The number of terms in the simplified expression equals the value K that we wish to compute.

3. EXPERIMENTAL RESULTS

While SocialFusion has not been fully implemented, we have begun evaluating various components of the system.

3.1 SocialFlicks: A SocialFusion Application

SocialFlicks is a SocialFusion application that displays recommended movie trailers that match the movie preferences of one or more users jointly watching a common display. This application could be deployed in locations such as video-rental establishments for marketing purposes, to make customers aware of new video rentals that match their interests. Like all applications that use the SocialFusion framework, SocialFlicks consists of stationary and mobile components. The stationary components were prototyped using the Java Standard Edition (SE) 5.0 platform, while the mobile components were prototyped using the Java Micro Edition (ME) platform. The mobile components were evaluated using the MicroEmulator tool, which is a pure Java implementation of Java ME in Java SE. All interactions between the stationary and mobile components are managed using Bluetooth. The BlueCove JSR-82 Emulator tool was used to emulate all Bluetooth connectivity.

3.1.1 Mobile Component

The mobile component (MC) in SocialFlicks shares a user's Facebook ID with the stationary component (SC) using Bluetooth. The MC user interface presents the Facebook login screen to the user on the user's mobile phone. The MC component assumes Internet access over a WWAN or wireless local area network (WLAN) connection so that the Facebook Web site can be accessed. After the user enters his/her Facebook user name and password, the MC logs into Facebook and obtains the user's Facebook ID using the Facebook REST API Web service. The MC then proceeds to launch a SocialFlicks Bluetooth service that enables the user's Facebook ID to be shared with the SC.

3.1.2 Stationary Component

The Stationary Component (SC) in SocialFlicks detects the presence of users watching a single common screen managed by the system and shows recommended movie trailers based on the favorite movie preferences expressed on each user's Facebook profile. The SC begins by searching for available mobile devices running the SocialFlicks Bluetooth service. Since Bluetooth radios on mobile phones are Bluetooth Class 2 devices with an approximate range of 10 meters, only nearby users will be detected. After detecting an appropriate mobile device, we query the device to determine the user's Facebook ID. In the current implementation, the user's Facebook ID is encoded as the Bluetooth service name of the SocialFlicks Bluetooth service running on the mobile device.

After obtaining the Facebook ID of each detected user, the SC constructs a playlist of recommended movie trailers for each user. First the SC uses the Facebook REST API Web service to obtain the contents of the "favorite movies" field on the user's Facebook profile. Next, the SC uses the Netflix REST API Web service to obtain a list of similar movies for each of the user's favorite movies. We assume that the user will prefer to watch movies that are similar to his/her favorite movies. If just one user was detected by the SC, then a single-user playlist generation heuristic can be applied to generate the final movie trailer playlist. For the SocialFlicks prototype, the first five recommended movies for each of the user's favorite movie titles are concatenated into a single aggregate list, and this aggregate list is sorted by movie release date. If multiple users were detected, then a group playlist generation heuristic can be applied to generate the playlist. For this prototype, we implement a simple heuristic where the top three most highly rated movies from each user's playlist are concatenated to form a playlist for the group. After generating the playlist, the SC searches for

Figure 3: Screenshot of prototypes of the SocialFlicks mobile and stationary components

corresponding movie trailers that are availabel on the local filesystem. Then, the SC begins playing the trailers on the screen using the VLC media player.

The SC performs the search for available mobile devices every ten seconds, although this search frequency can be adjusted. The current list of detected users is compared with the previous list of detected users. If these two lists are the same, then SocialFlicks continues playing movie trailers from the previously generated playlist. If the current list is different from the previous list, then SocialFlicks has detected that the group of users watching the screen has changed, and a new playlist is generated for the current group of users. Several options are being investigated for determining when to begin playing trailers from the new playlist. One option is to wait for the current movie trailer file to finish playing before beginning to play trailers from the new playlist. This would prevent abrupt changes to trailer playback when there is a lot of user movement by establishing some hysteresis.

Figure 3 shows a screenshot of the current prototypes of the SocialFlicks MC and SC. The MC for one user is running in the MicroEmulator window, while the SC has generated a movie trailer playlist for this user and invoked VLC to begin playing trailers from this playlist. VLC is playing the trailer for the film "Minority Report".

3.2 Implementing K-anonymity

To test our approach's feasibility, we have applied Boolean minimization to calculate K over a small set of Facebook profiles consisting of 230 users, using only their preference data. Our results show that the proposed technique is efficient and accurate for computing K over this small sample. Typical values of K computed were 20 or less over this data set. The results were obtained in near real time. However, this initial feasibility test does not incorporate friendship data contained in social graphs, nor does it integrate heterogeneous data streams such as location histories, calendars, or sensor data. Moreover, how this approach scales to large numbers of users was not evaluated.

4. CONCLUSIONS

We have presented SocialFusion, a context-aware system for fusing mobile, social, and sensor data streams to effectuate a context- aware action both for an individual and for groups of people. We have outlined a multi-stage architecture for achieving such fusion, and described some of the major challenges encountered: collecting and managing diverse data streams; mining the data for context-aware inferences of individuals and groups; generating context-aware recommendations; and preserving privacy and anonymity. We have demonstrated an early context-aware video application of SocialFusion called SocialFlicks.

5. REFERENCES

[1] M. Weiser, "Some computer science issues in ubiquitous computing," *Communications of the ACM*, vol. 36, no. 7, 1993.

[2] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han, "Whozthat? evolving an ecosystem for context-aware mobile social networks," *IEEE Network*, vol. 22, no. 4, 2008.

[3] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, 2005.

[4] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proc. of ACM SenSys*, 2008.

[5] C. M. Gartrell, "Socialaware: Context-aware multimedia presentation via mobile social networks," Master's thesis, University of Colorado at Boulder, December 2008.

[6] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme - injecting sensing presence into social networking applications," in *Proc.of EuroSSC*, 2007.

[7] T. Kindberg and J. Barton, "A web-based nomadic computing system," *Computer Networks*, vol. 35, no. 4, 2001.

[8] W. K. Edwards and R. Grinter, "At home with ubiquitous computing: Seven challenges," in *Proceedings of ACM Ubicomp*, 2001.

[9] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, "Developing a context-aware electronic tourist guide: some issues and experiences," in *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2000, pp. 17–24.

[10] M. Wirz, D. Roggen, and G. Tröster, "Decentralized detection of group formations from wearable acceleration sensors," in *Proc. of IEEE SocialCom*, 2009.

[11] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," *Proc. VLDB Endow.*, vol. 1, no. 1, 2008.

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 6, 2005.

[13] A. Jameson, "More than the sum of its members: challenges for group recommender systems," in *Proc. of ACM AVI*, 2004.

[14] G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," University of Massachusetts Amherst, Tech. Rep., 2007.

[15] Q. Wei and Y. Lu, "Preservation of privacy in publishing social network data," in *Proc. of IEEE ISECS*, 2008.

[16] B. Thompson and D. Yao, "The union-split algorithm and cluster-based anonymization of social networks," in *Proc. of ACM ASIACCS*, 2009.

[17] A. Campan and T. Truta, "A clustering approach for data and structural anonymity in social networks," in *Proc. of ACM PinKDD*, 2008.

[18] R. K. Brayton, A. L. Sangiovanni-Vincentelli, C. T. McMullen, and G. D. Hachtel, *Logic Minimization Algorithms for VLSI Synthesis.* Kluwer Academic Publishers, 1984.

[19] "Marge," https://marge.dev.java.net.