Building an Information System for Reality Mining Based on Communication Traces

Matthias Steinbauer
Johannes Kepler University Linz
Department of Telecooperation
Linz, Austria
Email: matthias.steinbauer@jku.at

Gabriele Kotsis
Johannes Kepler University Linz
Department of Telecooperation
Linz, Austria
Email: gabriele.kotsis@jku.at

***Abstract*—Reality mining (RM), i.e. gathering and analysing data about human behaviour and interaction in the real world, is gaining increasing interest in various disciplines. In this paper, we are discussing RM from a computer science perspective. The objective is to propose a reference architecture supporting the technical issues associated with RM. We argue for a web service based architecture providing the necessary performance, flexibility, extensibility and interoperability required in RM.**

A prototypical implementation is presented, demonstrating how sensor data from communication and location sensors of a group are being sent to a web service for further analysis. The gathered information can be used, for example, to analyse social networks in real time. In our example application, we calculate scores for each individual group members behavioral stereotype, which provides information for detecting bottlenecks within group communication. This information is then reported back, following the idea of sociometry, which reflects the groups current situation back to the group in order to allow improvements.

Feedback and reporting are another important functional requirement in RM. In our prototype, we provide such functionality in software. The implementation as a web service allows easy integration into existing information and communication systems.

I. Introduction

During every social interaction, humans find themselves listening heavily to social signals outside the content of a conversation [19]. While transporting and analysing conversation is a quite common task for machines, the transport or analysis of social signals is not. As part of an evolving research field called Reality Mining, machines learn to extract social clues from social systems [8], [23].

This is done by applying algorithms known from the field of Data Mining to real-time data. Sensors are able to record conversations, movements and the activity state of individuals. These signals are utilized to better target advertisements, guide traffic, improve human health or to optimize group processes [15].

In Reality Mining, a sensor could be defined as any source of data, which is human generated. Often sensors are chosen such that they are already deployed in the environment, which is to be sensed. This means that a smart phone's microphones could be the sensor of an audio based Reality Mining system. Other examples of sensors are location traces from GPS enabled devices, proximity data from Bluetooth signals or communication traces, which are left from phone calls or text based messaging.

In this work, we propose a system, which is able to collect communication traces from mobile devices. These traces are used to create a model of the monitored social network. This model then, in turn, should form a basis for further work, which is to create Reality Mining applications that use this social network model. We will show that this system is able to record communication traces from various sources of communication. This could be in the form of phone calls, e-mail messages or other text based messages. Further, we will integrate information gathered from Bluetooth proximity sensing into the same model in order to record face-to-face meetings in just the same way as any other form of communication.

This system is modeled as a client server architecture in order to easily attach new sensors to the system as the need arises. This paper focuses on the creation of such a system, which we intend to use in further work to detect behavioral stereotypes in workgroups, as defined by Schindler [21]. We think that such behavioral stereotypes are identifiable solely by running data mining algorithms on the data set of communication traces of a work group.

Within this work, we will define a social network as a model, which is modeled as a directed graph. It consists of nodes for each individual in a group and each single occurrence of a communication interaction will be modeled as a directed edge [25] [26].

In section II, we briefly present work from the field of Reality Mining. Section III explains requirements we identified for such a system. In IV, we explain our current prototype implementation and how it works. The reader will find a brief outlook on our future work in V and, finally, concluding remarks in VI.

II. Related Work

The field of Reality Mining describes new ways of sensing complex social systems in real time [8], [22]. Sensors are plentifully available and may already be deployed in our everyday lives. Our mobile phones and computational traces provide a large amount of valuable data for real time Reality Mining.

Waber and Pentland claim [24] that analysis of organizations in real time is a new topic. Current organizational analysis has always used surveys to discover the latent structure of an organization. It references a way of becoming aware of social structures in real time. In their work, a little hardware badge is used to determine spacial proximity. As the badge works with infrared light, it is also capable of determining if two communication partners are facing each other. Using the data gathered by those badges, the system is capable of creating a graph of the social network, which is built during everyday contact. Voice recording capabilities allow for the checking of whether or not a person is speaking. Those badges are called Sociometric badges, since they are capable of creating metrics about social interaction.

Interaction Process Analysis is a technique used to describe communications processes [7] as well as a technique used to analyse group discussions and interactions. This analysis may also be used to analyse the way a group is solving a problem. Traditionally, this process is very time consuming, since human observers have to mark events and write logs about social interactions. Dong and Pentland present an idea to automate this with Sociometric badges and describe how a communication model can be created. With stochastic methods, they are able to analyze dynamics and performance in observed groups.

A more sophisticated idea is to not only analyze isolated discussions, but monitor a group for a complete work day and gain a model of their social network [18]. By correlating this model with a variety of organisational relevant outcomes, such as performance and job satisfaction, one may become aware of social network patterns, which result in organisational outcomes. The proposition is to be able to predict organisational outcomes by looking at social network characteristics.

The briefly presented work up until now focused on groups and their behaviour. Other applications show that Reality Mining may be used to help avoid traffic congestions. Creating a web of mobile sensors that are placed in users cars, which are constantly reporting their current GPS location back to a central server, it is possible to detect places, which are in danger of traffic congestion. Users can use such a system in order to avoid those congestions and use detours, which are proposed by the system [14].

A further area where Reality Mining could be exploited is the improvement of human health. As an example, Ginsberg et al. showed that the analysis of search engine queries can be used to follow influenza epidemics. This information is useful for health professionals to respond better to seasonal epidemics [10].

Lastly, advertising is an area where Reality Mining could potentially contribute. By applying data mining methods on social network models, groups may be identified. If those groups share some common interest, targeted advertising may be far more effective than broadcasting [27].

Our work will mainly focus on communication traces, which are left on users smart phones and computers. In contrast to the work on group analysis presented in this section, our work is not using dedicated sensor hardware. This way, we eliminate the need to attach the sensor hardware to a computer from time to time in order to download data. Our system is targeted to be permanently online and to be able to provide feedback to users in real time. The system focuses on sensor data, which is already generated by computing devices. Planned applications are targeted towards group analysis.

III. Requirements

Section II showed very different application areas of Reality Mining. Specifically, our work is focused on group processes. We want to be able to reason about group performance and behaviour upon a groups communication patterns. The following will present requirements, which are derived from our fields of interests. These are the detection of different types of groups within a larger social network model and the detection of behavioural stereotypes of individuals in these groups. Further, we intend to use the system in order to improve group communication processes and workflows.

A. Architecture

We assume that a client server architecture is most feasible for such a system. Regarding privacy concerns, one will more likely be willing to share communication traces with a common centralized system than using a peer-to-peer based system, which constantly reports sensitive data to all participating users. Since sensors are meant to run on many different devices, like mobile phones, computers or even on dedicated sensor hardware, a simple way of sending sensor results to a server component is needed. This transmission needs to be encrypted and clients need to be authenticated to the server in order to prevent fraud data from being committed to the system.

B. Sensors

Current planned sensors are monitors that collect data about phone calls, e-mail communication, text messages and Bluetooth proximity. Since further sensors may be added at a later time, the data models and the sensor interface need to be extensible in such a way that new types of sensors may be added to the system without the need to rebuild the complete software stack from server to clients.

Sensors will concentrate on communication and will be implemented in many different ways, like on mobile devices, computers or on dedicated sensor hardware. In order to create robust system sensors, one should not rely on permanent network connectivity. Sensor data needs to be backlogged for later delivery.

To address these requirements, a loosely coupled interface is to be created, which uses a simple communication contract to be implemented by clients who send sensor data. This contract needs to be able to accept log records, which contain information about the initiator and the target audience of a communication record. Further, the type of sensor and a timestamp need to be specified for each record.

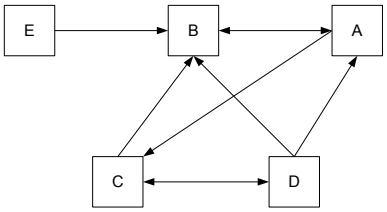

Fig. 1. Simple example of the proposed model, showing five nodes for each of five persons in a work group. The edges in the diagram refer to communication relations between individuals.

For later use, log records should also be able to carry some type specific additional data. This could be the time of a phone call or the subject of e-mail messages. The later, for instance, may be used to keep track of e-mail message threads.

C. Model

Our social network should be modeled like the example in figure 1, which is closely drawn after sociograms, as defined by Moreno [16]. It uses nodes for each person who contributes data to this social network. The edges show relations, which are made up of communications. The data model stores a single edge for each occurrence of any two group members communicating, which is displayed as a weighted edge in visualizations.

The data model needs to cope with the fact that different types of sensors are used to create the social network model. It needs to implement a way of distinguishing between the different types of communication occurrences, which are integrated in the model. This might be implemented with a type field, for instance, to discriminate e-mail and phone communication.

D. Grouping and Timeframes

The model proposed above will create a detailed view of the communication network of the participating users. This will result in large amounts of data over time, which will need some basic filtering before processing. Two filters, which will be required for many use cases, will be a group filter and a time frame filter.

The group filter will allow an application to consider only communication traces from a certain group of individuals. These groups may be defined by a user or by an algorithm, but the filter has to make sure that only traces, which originate from a person of this group and are targeted to a person from this group, are considered in later evaluation.

A second filter is the timeframe filter. Since groups run through several stages of group forming again and again [2], the social network model has to also adapt to this new situation constantly. In order to have algorithms adapt to the new social network model, they may simply abandon old log records that no longer relate to reality. This should be implemented with a filter that selects log data only from a certain time frame out of the complete set of available log data.

E. Privacy by Design

Since Reality Mining systems are designed to process highly sensitive data, we propose making privacy one of the design goals [13]. A worst case scenario from a privacy point of view is having all the communication log data become available to the public. To keep a users data as secure as possible, the system should store only data, which is crucial for our needs. This means that, unless the necessity arises, no communication content is stored.

Further, the system has to make sure that only registered and authorized users are able to access the communication model. A user should only be able to see data in detail, which was generated by them.

Sensors will use an interface with a contract that only allows posting of new records, but does not provide any means of data retrieval or manipulation. Data access is only to be granted to code that is running within the server.

IV. Prototype Implementation

In order to prove the concept, a prototype has been created[1]. It shows how sensor data from communication and proximity sensors are being sent to a web service for further analysis. It consists of several parts, including a service component, which is provided as a SOAP service in order to provide an interface, which clients may use to log sensor data.

A. Service

SOAP, as specified by the W3C [9], was used here, as new clients eventually need to be added on a regular basis and web services provide a good foundation for standardized communication. When using web services, there is no need to create an implementation of the communication protocol for each new platform. Platforms implementing SOAP based web services usually provide a means of creating an access layer from a web service definition provided in WSDL [5].

Also, the relatively loose coupling of web services compared to purely stream based socket communication is a huge advantage. As sensor data is mostly originating from mobile devices with connection quality that may vary, web services provide a good service, as socket connections are only created on demand and some basic error handling is already available due to the HTTP protocol. As an alternative, RESTful [20] web services were investigated, but they were found to be too loose in terms of not providing a contract between client and server. Further, providing a RESTful service may suggest to the consumer of that service that all default functionality, which is expected by a RESTful service, is available within this system. Many RESTful web services provide access to an object via a URL and allow some default operations on those objects in order to retrieve and manipulate data. Included in this group of operations are HTTP methods PUT, POST, GET and DELETE [17]. This is absolutely not the case for this prototype, which is only capable of accepting log data via its SOAP service and does not provide any interface in order to further read or manipulate data.

The service provides simple methods for logging communication data to the system. Most clients will use a method that

[1]The source code is available from http://www.steinbauer.org/publish/aa-code.zip

stores a log record of arbitrary type. The client needs to specify a sender and a target of each communication occurrence. Any of a users phone numbers, e-mail adresses or Bluetooth MAC adresses may be used for this. As long as this data is recorded within some users profile, the system will be able to match the data up correctly within the model. Further, the type of log record needs to be specified. Optional fields are a timestamp and binary field for miscellaneous data, which is type specific. The timestamp is needed if a client sends data, which was sensed earlier. Miscellaneous data may, for instance, be the subject field of an e-mail message.

B. Clients

In order to fill the service component with data, several clients were created. For desktop computing, a client was created, which is capable of monitoring the users outgoing e-mail traffic. This is done by providing an SMTP proxy service such that the user is able to reconfigure his or her e-mail client to use the locally running proxy. The proxy then in turn connects to the users SMTP server for e-mail delivery. For each message that is delivered via this proxy, the web service is also notified of the sender and recipients e-mail addresses, the sending timestamp of the e-mail and the subject.

For the mobile device sector, the Android platform was chosen as the first client platform. This is the case because the Android SDK allows access to all needed data if an app asks for permissions properly. The mobile client is used to collect phone call and Bluetooth proximity data. It runs as an application and a service within the Android ecosystem. The application is merely a configuration interface, which the user may use to configure when and how his or her data is sent to the server. The user may decide if the devices Bluetooth proximity sensing is turned on, if detected log records may be sent to the server automatically and the user may specify how often the background service is launched in order to check for new phone calls in the handsets call log.

If Bluetooth proximity sensing is turned on, the service runs every $30s$ in order to scan for other Bluetooth devices. The service then keeps a list of nearby devices. If one of these devices is removed from this list, because it left the proximity range, a proximity record is recorded and sent to the server.

For every record that is sent to the server, a notification on the Android information screen is generated. If the user enabled automatic upload of personal data, this is merely an indication that new data was sent to the server. Otherwise, the user may select this notification and decide if the data should be transmitted.

We have also created a client, which is capable of batch importing e-mail messages stored in MBOX formatted local mailboxes. This client was used to import the Enron [1] e-mail database for test runs during development.

C. Web-based Interface

Since the clients, which are available for data logging, do not provide any interface to access the data, there is a need for another front end, which allows users to log into the system and view the social network model and results from algorithms that were run on that model.

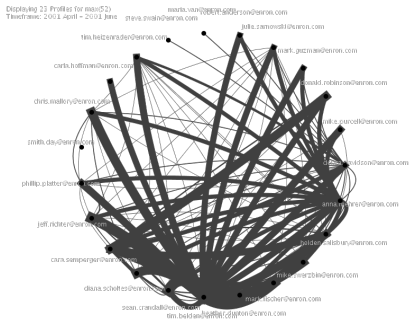

Fig. 2. Visualization of an e-mail based data set imported via bulk import. The e-mail data is from an Enron corporation data set. The figure shows data from the 1st of April 2001 to the 30th of June 2001

In our system, a core component is a visualization of the stored social network model, as shown in figure 2. It shows a dot for each person in the model, labeled with the users profile name. They are lined up in a circle around the models center. Edges between profiles are represented with curved lines. They bend so that edges in one direction do not cover their reverse counterparts. The edges thickness depends on the number of connections that are represented by one single edge. While the logical model contains a log record for each real connection that was logged, the visualization shows, at most, one edge for each direction between any two nodes.

The current visualization model also allows the application of filters for the type of record to be displayed. This allows the creation of social network models, which display, for instance, only e-mail communication data or only Bluetooth proximity data. This way users are enabled to quickly view the different communication patterns that have occurred on different types of communication channels.

V. Future Work

Having created an implementation of this proposed framework, we are planning to use it to study work groups. We are especially interested in group interaction and group performance.

In the first project, we intend to use our prototype implementation to detect behavioural stereotypes for each individual group member, as defined by Schindler [21]. The social network model, which is created by the system, is to be interpreted as a sociogram [16]. Upon this sociogram, several algorithms will be implemented and put to the test in order to assign a behavioural stereotype to each member of a work group.

A further field of research will be the detection of bottlenecks within a social network, which are defined as people that are so central in a social network, that they end up holding the group back if they cannot keep up with their workload [3]. Since the social network modeled within our system is focused on communication traces, it is also usable for the

optimization of communication processes or workflows. Communication bottlenecks might be spots in the social network where significant high loads of communication have to be processed. Also, it is important to a groups performance to identify highly peripheral people within the communication network. They may represent under utilized resources for the group and should get reconnected to their group [4].

In Requirements Engineering, we intend to use this system to support the development of CSCW applications [11]. Again, bottlenecks should be identified and adressed by future CSCW systems. Especially in places where one artefact is sent from one group member to the other and stalls in this forwarding could delay the groups progress. Further groups and their main communication needs can be identified and, as a result, support the development of CSCW applications.

VI. Lessons Learned

With the prototypical implementation presented in this paper, we were able to demonstrate that the proposed system can collect and integrate data from various existing data sources. Our current implementation includes the tracking of phone calls, e-mail traffic and Bluetooth proximity. Other types of sensors can be easily integrated. The collected information is sent to a central server, but could alternatively also be processed locally depending on the application needs.

The current implementation also shows that current state-of-the-art mobile and web service technology meets the RM requirements.

In a small demo application, we have demonstrated how the collected data allows insights into the social network model that was created from the recorded data. On one hand, we are able to create model visualizations and are able to experiment with different forms of visualization to find graphical representations that fit different application areas. On the other hand, we are able to experiment with algorithms, which exploit this data in order to obtain social clues about social systems.

Our system is designed to automatically collect data and to continuously integrate new sensor data into the social network model. This way, the model and the derived information is constantly kept up to date with the real world.

From a real time point of view, experiments, which imported the e-mail database from Enron [1], showed that a relational database model may not be sufficient for large scale use of this system. The social network model created from that dataset resulted in a PostgreSQL database of 411MB in size. Queries based solely on the table, which holds the log records, took up to 7s. The application then needs up to 5 minutes to update the memory model and visualization of the social network model from a group of 25 profiles.

Since datasets collected in larger scenarios will perform even worse, we will have to rethink the programming model and base further evolutions of our system on technology, which is capable of handling Big Data [12] and uses a Map Reduce programming model [6].

References

[1] William W. Cohen. Enron email dataset. http://www.cs.cmu.edu/~enron/, 08 2009.

[2] David Coleman and Stewart Levine. *Collaboration 2.0*. HappyAbout.info, 2008.

[3] Rob Cross and Andrew Parker. *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*. Harvard Business School Press, 2004.

[4] Rob Cross, Andrew Parker, and Stephen P. Borgatti. A bird's-eye view: Using social network analysis to improve knowledge creation and sharing. Technical report, IBM Institute for Knowledge-Based Organizations, 2002.

[5] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing Magazine*, pages 86–93, 2002.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[7] Wen Dong and Alex (Sandy) Pentland. Quantifying group problem solving with stochastic analysis. In *International Conference on Multimodal Interfaces*, 2010.

[8] Nathan Eagle and Alex Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[9] Martin Gudgin et al. Soap version 1.2 part 1: Messaging framework (second edition). http://www.w3.org/TR/soap12-part1/, 2007.

[10] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009.

[11] Jonathan Grudin. Why cscw applications fail: Problems in the design and evaluation of organizational interfaces. In *ACM conference on Computer-supported cooperative work*, pages 85–93, 1988.

[12] Adam Jacobs. The pathologies of big data. *Communications of the ACM*, 52(8), Aug. 2009.

[13] Marc Langheinrich. Privacy by design – principles of privacy-aware ubiquitous systems. In *Ubicomp*, pages 273–291, 2001.

[14] Chia-Hao Lo, Wen-Chih Peng, Ting-Yu Lin, and Chun-Shuo Lin. Carweb: A traffic data collection platform. In *The Ninth International Conference on Mobile Data Management*, pages 221–222, 2008.

[15] Tom M. Mitchel. Mining our reality. *Science*, 326:1644–1645, 2009.

[16] Jakob Lucas Moreno. *Who Shall Survive*. Beacon House Inc., 1934.

[17] Michael Muehlena, Jeffrey V. Nickersona, and Keith D. Swenson. Developing web services choreography standards—the case of rest vs. soap. *Decision Support Systems*, 40:9–29, 2005.

[18] Daniel Olguin Olguin and Alex (Sandy) Pentland. Sensor-based organisational design and engineering. *Int. J. Organisational Design and Engineering*, 1(1):69–97, 2010.

[19] Alex Pendland and Trac Heibeck. *Honest Signals, How They Shape Our World*. The MIT Press, 2008.

[20] Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly Media, 2007.

[21] Raoul Schindler. Grundprinzipien der psychodynamik in der gruppe. In *Psyche*, 1957.

[22] Sarah Underwood. Making sense of real-time behavior. *Commun. ACM*, 53(8):17–18, 2010.

[23] Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard, and Alex Pentland. Social signal processing: State-of-the-art and future perspectives of an emerging domain. In *ACM Multimedia*, pages 1061–1070, 2008.

[24] Benjamin N. Waber and Alex (Sandy) Pentland. Augmented social reality. Technical report, MIT Media Laboratory, Human Dynamics Group, 2007.

[25] S. Wasserman, K. Faust, D. Iacobucci, and M. Granovetter. Social network analysis: Methods and applications. *Cambridge University Press*, 1994.

[26] B. Wellman. For a social network analysis of computer networks: A sociological perspective on collaborative work and virtual community. In *The ACM SIGCPR/SIGMIS conference*, pages 11–13, 1996.

[27] Wan-Shiou Yang, Jia-Ben Dia, Hung-Chi Cheng, and Hsing-Tzu Lin. Mining social networks for targeted advertising. In *39th Hawaii International Conference on System Sciences*, 2006.