CoDaMine: Communication Data Mining for Feedback and Control in Ubiquitous Environments

Tom Gross, Mirko Fetter
Faculty of Media
Bauhaus-University Weimar
Bauhausstr. 11
99423 Weimar, Germany
<firstname.lastname>(at)medien.uni-weimar.de

Abstract

In ubiquitous environments an increasing number of sensors capture information on users and an increasing number of actuators present information to users. In this paper we present CoDaMine—a novel approach for providing users with system-generated feedback and control in ubiquitous environments giving them the freedom they need while reducing their effort. Basically, CoDaMine captures and classifies users' online communication to learn about their social relationships in order to provide them with recommendations for inter-personal privacy and trust management.

1 Introduction

In ubiquitous environments an increasing number of sensors capture information on users and at the same time an increasing number of actuators are available to present information to users. This vast capturing of information potentially enables the system to be well informed about users and to consequently, quickly adapt to the users and provide them with the information and changes to the environment as they need it.

At the same time users can encounter new challenges. In particular, the system can violate the users' privacy by capturing information that the users do not want to share, and the system might disrupt the users by being too obtrusive in its information supply or adaptation. There are many reasons for the particular social implications of ubiquitous computing concerning privacy and trust; Langheinrich [17] names the following: ubiquity (i.e., the fact that ubiquitous computing aims to be available anywhere); invisibility (i.e., the fact that ubiquitous computing aims to be calm and to disappear); sensing (i.e., the fact that ubiquitous computing means for sensing data get increasingly smaller and can capture an increasing number of fine granular information); and memory amplification (i.e., the fact that ubiquitous computing technology often continuously captures data).

The concept of faces has been used as an approach to resolve this dichotomy—maintaining the advantages of capturing data and informing users, while preserving privacy and minimising disruption. According to Goffman [7] humans construct faces or social identities that represent a subset of characteristics and information on them and that are revealed to specific audiences. We have developed a concept and system for such a selective information disclosure in the context of instant messaging, where users can create faces that contain other users as well as information that they want to share with these users [8].

In complex ubiquitous environments maintaining an overview of one's faces including the respective users and information as well as managing and keeping one's faces up-to-date can become a challenge and a considerable effort for the user. Yet, as Bellotti and Sellen [3, p. 77] point out in their 'framework for design for privacy in ubiquitous computing environments' users need 'feedback' and 'control':

- *Feedback* provides users with information on the data that are captured about them
- *Control* allows users to specify personal preferences on this capturing and sharing of information

In this paper we present CoDaMine—a novel approach for providing users with *system-generated feedback and control* in ubiquitous environments giving them the freedom they need while reducing their effort. Basically, CoDaMine captures and analyses the users' online communication, and thereby learns about the users' conversations and social relationships with other users. It can then, based on this knowledge, make recommendations for users' configurations of faces in ubiquitous environments. So, overall the users get feedback on their current specifications for information sharing and recommendation on the control of useful future configurations.

The paper is structured as follows. We present the concept and implementation of CoDaMine. We give a brief overview of related work. And, finally, we summarise the paper in the conclusions and glance at future work.

2 CoDaMine Concept

The concept of CoDaMine basically departs from a perspective that users of ubiquitous environments are able and willing to manage their privacy and trust by controlling the capturing of their data. While this is not true for any and all circumstances, there are many examples where users have this power (e.g., in their private homes, in their personal work offices).

Privacy can be seen from various perspectives with many resulting definitions [19]. In this paper with privacy we mean the ‘concept of controlling the dissemination and use of one’s personal information’ [16]. There is no generally accepted definition of trust. Many authors point out that trust is the expectation that somebody else has power and the belief that this person will not use this power to harm us. In this paper the important aspect of trust is that ‘it allows us to reveal vulnerable parts of ourselves to others’ [6]. Advanced management of privacy and trust allows users to have multiple privacy and trust settings depending on the context and social setting.

We already developed a concept and system for advanced management and trust for *instant messaging* in *PRIMIFaces* [10]. PRIMIFaces has its origin in Goffman’s concept of faces [7] grounded in sociology and psychology and has been transferred to the field of presence and awareness in instant messaging. A face according to Goffman defines a specific front that a person shows in a specific setting to a specific audience. A face in PRIMIFaces translates into the presentation of the self, mapped to information the user wants to disclose to a particular group of people in a specific online situation. As a result the user specific configuration of a PRIMIFaces instance, with its different face names and the assigned contacts and information sources together form an image of the different social contexts of this user. Like in the real world, faces in PRIMIFaces are not static, but can evolve over time. This means that users constantly have to adapt their configurations. For instance, as a person starts to become friends with a working colleague the privacy and trust settings need to be adapted for this person. In the existing PRIMIFaces this can be very time-consuming and tedious.

The concept of *CoDaMine* presented in this paper aims to support the lightweight management of trust and privacy over time by both allowing users to manually specify faces, and contacts, and information sources, and providing users with system-generated suggestions for adaptations to their trust and privacy settings over time. Suggestions for adaptations are based on the messages exchanged between two online users. Our approach is to analyse the linguistic features of those messages and how they correlate to the different faces of a user. This approach is rooted in two central findings: speech communities in sociolinguistics, and conversation contents properties in text-based computer-mediated communication.

Speech communities in sociolinguistics—historically and conceptually discussed by [22]—is a concept of group members communicating with each other for a special purpose and using language in a specific, unique, and mutual way. These communities can also be found in online communication and form specific linguistic practices [21, 25]. Often these groups share the same topics and therefore develop a common, specialised vocabulary, characteristic terminology or idiom—a jargon. For instance, the project members of an IT project might frequently use terms for technologies and tools (e.g., Java, or XML-RPC).

Conversation content in text-based computer-mediated communication has specific properties. In face-to-face communication a person adapts its speech, mimic, and gesture to the situation and audience. According to [24] and [4] and [2] these mechanisms in order to ‘save face’ can also be found in text-based computer-mediated communication and may be an indicator for the current face. Due to the lack of other communication channels and the missing physical context the language is augmented with features that mimic the spoken language (e.g., ‘Ahem…’, ‘GREAT!’), imitate auditory information (e.g., ‘*sniff*’), or represent facial expressions or physical actions (e.g., ’(hug)’) [11]. The level of formality and complexity of the vocabulary, the use of abbreviations (e.g., ‘dunno’, ‘ROFL’, ‘cu’), or emoticons (e.g., ‘:)’, ’:-o’) often reflects the social and situational context of the conversation. Therefore, this can be seen as face-work, which is often encountered especially in instant messaging, where strong ties between the users, and the transient nature between written and spoken dialogue facilitate this kind of linguistic mechanisms [26].

The analysis of linguistic features in online communication as well as their correlation to sociological variables has mostly been studied in a post-hoc manner through ethnographic approaches like observations, analysis of log-files, or interviews. As these linguistic peculiarities manifest in a machine readable form in the message history of each communicating dyad exchanging text messages, our approach is to automatically analyse these specific features in real-time by applying text analysis and text mining techniques and to use these information to allow a better privacy and trust management in ubiquitous environments.

3 CoDaMine Scenario

In this section we present a scenario explaining from the user’s point of view how CoDaMine how users interact with the system, create messages, and are informed about recommendations.

In order to manage privacy and trust users start their CoDaMine client. As users begin using CoDaMine they start to map their social bindings, contexts and information to the application. They create new faces that match their specific social contexts. Users add any number of contacts that belong to the respective face, and information sources that are visible to the contacts included. So, users establish an information flow between the faces' information sources and the assigned contacts.

Once users have a satisfying initial configuration, they can start communicating with other users represented by the contacts in their faces. They initiate each conversation by selecting a contact in the face that most closely matches the context of the intended communication.

During online text communication the system analyses the exchanged messages in the background. Thereby, the system detects whether the contents of the messages are suitable for the selected face, or whether alternatively the messages sent would better fit to another existing or new face. A dialog box presents the resulting recommendation to the users and allows them to accept or reject recommendations. The following example shows the impact of the user's decision. Klaus, a student, has the contact Julian in his face 'University'. They already had several online conversations during the semesters about the lectures and exams they had to take. In the next semester they both take part in a student project at their university. The system learns that the users have a face 'CoDaMine' and the contents of conversation in the context of this project. In the communication between Klaus and Julian the topics of their conversations evolve towards issues related to the project. So, the system suggests Klaus to add the contact Julian to the face 'CoDaMine' (Figure 1). If Klaus accepts the recommendation, CoDaMine adds the contact to the target face.

If Klaus had rejected the recommendation, the system would remember this decision and adapt its recommendation behaviour by generating future recommendations concerning Julian and the face 'CoDaMine' based on additional data.

4 Data Analysis

A central element of CoDaMine is the analysis of the communication data in order to find similarities and patterns that are adequate reasons for a recommendation to add or remove a contact from a face. The analysis of the data is done in several steps. Each step extracts further specific information that is required to infer the social coherences between the online communication partners.

4.1 Dealing with Asymmetrical Faces

Each analysed message belongs to two users: the sender and the receiver. Each of them has an individual view on the content of the communication data reflecting the context in which messages are sent and received.

As in CoDaMine faces can be asymmetrical—that is, a user can have another user in a certain face, whereas the second user has the first user in a different face—the face name of the respective user has to be taken into account as a parameter to analyse the message text from the user's point of view.

In order to provide users with automatically generated recommendations concerning the configuration of their faces a measuring unit is needed to determine changes in the communication. CoDaMine analyses the communication data on basis of each message sent.

We assume that each face contains a certain group of contacts, whose language use shows significant similarities. Therefore, it is possible to determine to which face a message fits best based on its linguistic particularities and keywords. The task of mapping a category to each message is a classification issue that can be addressed by applying data mining techniques.

4.2 Preparing the Data

In order to enhance the accuracy of the classification it is useful to prepare the data. The first step is to *remove stop words* from the text. Stop words have no significant meaning for the content, but for the structure of human language (e.g., 'a', 'of', 'the'). Since the conversation contents of online communication are rather informal, a large set is needed to reduce as much noise as possible. A further step is to reduce noise from the data by applying *stemming*. Stemming algorithms reduce words to their root. This decreases the diversity of the data that are processed by the classifier.

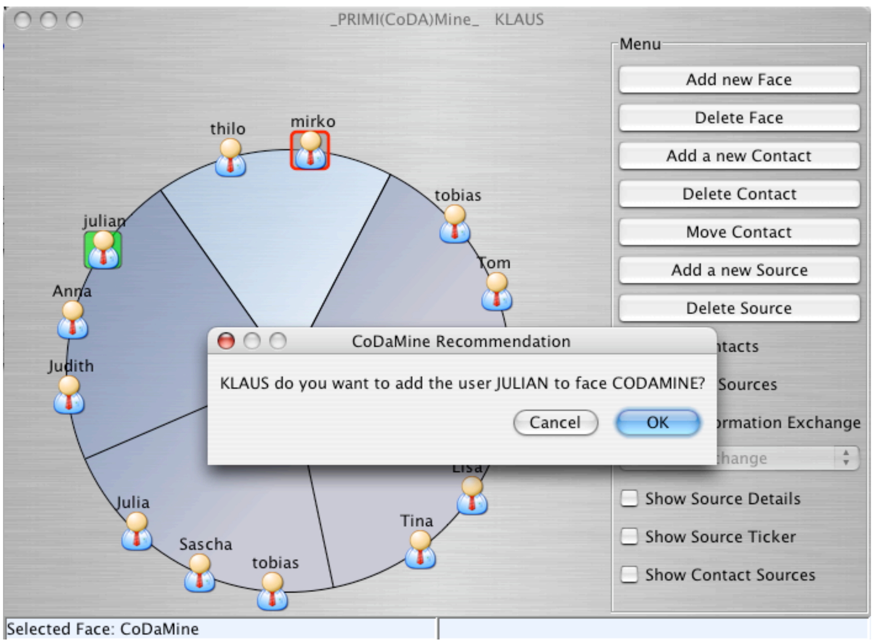

Figure 1. CoDaMine client with recommendation.

4.3 Training the System

Before the classifier can be used an adequate training corpus is needed. In CoDaMine the categories differ from user to user and it is not possible to provide initial training data. So, the data forming the corpus has to be collected dynamically, in the background during regular use of the CoDaMine system. The initial training period lasts until enough communication data (50 messages per user) is collected. These messages are manually pre-classified by the user to enable controlled training of a classifier—that is, users get regular messages and classify them before they answer them in their regular online communication. The effort for users is comparable with the maintenance of an e-mail spam filter. The users contribute by tagging conversations with the name of the face the conversation belongs to. This means that users initiate an online conversation by double clicking on a contact's icon in a specific face. When a sender is already sending messages to a specific recipient in the context of a specific face, then the sender can easily change the face to send the same recipient a message in a different face.

The number of faces can change over time. However, classifiers do not allow changes—that is, each time a user creates a new face, the complete classifier would have to be rebuilt. We use blank initial categories: for each user a classifier gets initiated with a fixed and sufficient number of free categories. With each new category a dictionary entry is created that maps the face name to a blank category that has to be trained.

4.4 Special Treatment

Furthermore, typographical signs and spelling errors get special treatment. *Typographical signs* such as emoticons and online communication specific abbreviations appear in online text conversations. We assume that the higher the number of smileys used, the more likely the communication is informal. Each user applies them in an individual manner. Therefore, this information is normalised per user for the process of recommendation creation. The purpose of analysing communication data on *spelling errors* is to reduce the weight of messages with misspelled words. For this purpose the spelling of each message is analysed, and the importance of the message for the classification is then weighted according to the proportion of misspelled words in the message.

5 Implementation

This section describes the implementation of CoDaMine. We outline its architecture and give a detailed description of the core parts—the inference engines.

5.1 Architecture

The system consists of three components. These are the CoDaMine client, the Wildfire instant messaging server, and the CoDaMine server (cf. Figure 2).

The CoDaMine components run distributedly and communicate via network. The CoDaMine client connects to the Wildfire Server via XMPP [15]. XML-RPC is used to connect the client with the sensor platform [1].

The *CoDaMine client* is implemented as a plug-in for PRIMIBase an open infrastructure for rapid development of instant messaging environment [9]. The CoDaMine plug-in is based on PRIMIFaces [10] and extends its functionality with two internal software sensors. The *MessageSensor* observes the communication data and sends each message the name of the corresponding face and the contact to the CoDaMine Server. The *RecommendationSensor* retrieves the processed data from the server.

As IM infrastructure a *Wildfire server* [14] is used. Wildfire is a freely available cross-platform real-time instant messaging server based on the XMPP.

The *CoDaMine Server* leverages the SensBase infrastructure [8]. SensBase is a sensor-based ubiquitous

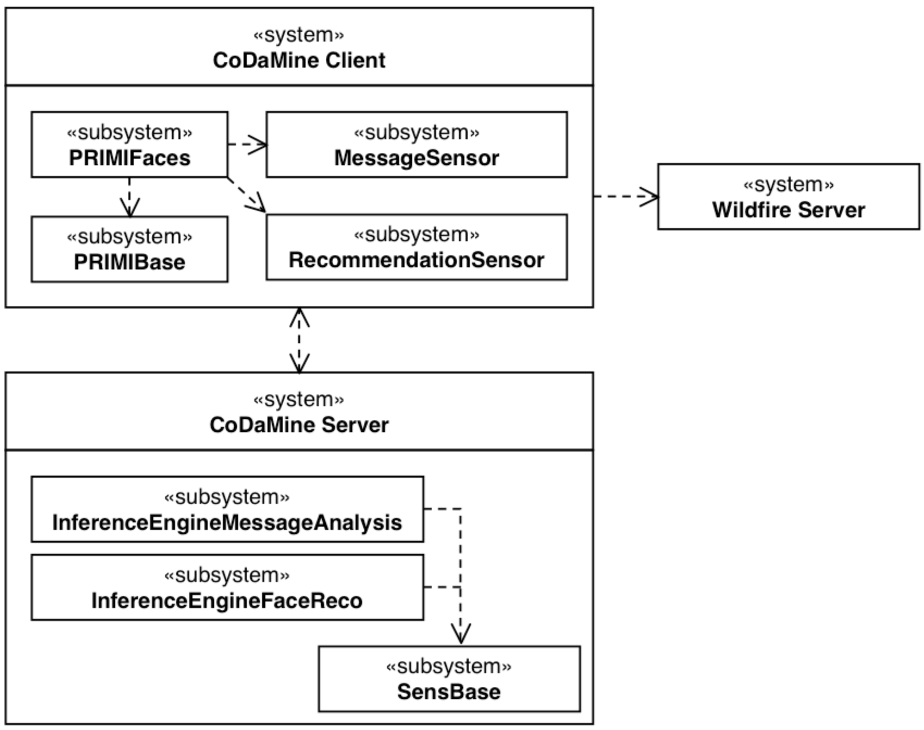

Figure 2. Component diagram of CoDaMine.

environment with a broad variety of sensors capturing real-world and electronic events and sending the event data via multifarious adapters to the SensBase server, and with a broad range of gateways for retrieving event data and presenting information to users. The inference engines are core part of CoDaMine and are described in detail below.

5.2 Inference Engines

SensBase and its inference engines support the server-side processing of sensor data. The inference engines provide mechanisms for easily integrating algorithms for processing sensor data via plug-in. In order to keep the inference engines simple, SensBase was extended by an interface that enables clients to remotely instantiate, register and configure a certain inference engine.

InferenceEngineMessageAnalysis

The *InferenceEngineMessageAnalysis* classifies users' communication data and determines the amount of spelling errors as well as the amount of typographical signs. Figure 3 shows the process chain of *InferenceEngineMessageAnalysis* as activity diagram.

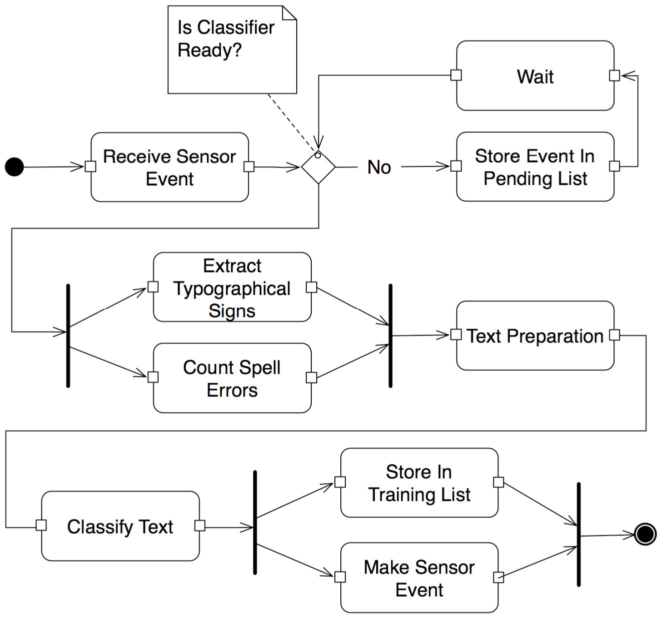

Figure 3. Activity diagram of the InferenceEngineMessageAnalysis.

The CoDaMine client sends a `SensorEvent` for each message a sender sends to a recipient containing name of the sender and the receiver, the message contents, and the face in which the message was sent to the CoDaMine server in the event's body.

In the CoDaMine server, the inference engine *InferenceEngineMessageAnalysis* checks if the classifier is ready. The classifier is ready, when enough training data has been entered, the training has finished, and the classifier is fully built. If the classifier is ready, the message is extracted out of the `SensorEvent.`

The typographical signs are extracted, and the spelling errors are counted. Then the inference engine prepares the text and classifies and stored it in the training list. It is used to permanently train the classifier in the background. The result of the processing is sent back as to SensBase as `SensorEvent`. If the classifier is not ready, the event is stored for later processing when the classifier is ready—that is, a thread sleep is executed and reacts on the change to the flag training-in-progress.

The system uses Weka for the text preparation and classification [27]. Weka provides state of the art implementations of algorithms for data mining. For the preparation of the text Weka's class `Stopwords` is used to find and remove stop words as well as the integrated Snowball stemmer to reduce the words to their root.

Weka's implementation of the Sequential Minimal Optimisation algorithm (SMO) [23] is used for training a Support Vector Machine (SVM) classifier. Research on a similar problem [12] showed that for short text data SVM is the best classifier.

As a basic exploration, we compared the performance of SMO for generating the SVM and C4.5 for generating a decision tree. The two algorithms were applied to a test corpus consisting of mailing list entries. The subsequent figures show the results of the comparisons of the algorithms accuracy (cf. Figure 4) and the time effort (cf. Figure 5).

The comparison of the accuracy showed that the SMO algorithm performs better than J48. The significant shorter time the SMO algorithm needed to build the model was the determining factor to choose this algorithm over the other for this specific problem.

The extraction of the number of spelling errors is done with the Jazzy Library [13]. The implemented filter uses the Jazzy dictionary to validate the spelling. The libraries can ignore emoticons—this is important, because otherwise typographical signs would be considered as errors in messages, which they are not.

The detection and counting of typographical signs is accomplished similar to the detection and counting of spelling errors. A dictionary is used to look up each token of a message. The results of the analysis are sent back to SensBase as `SensorEvent` and the subsequent inference engine *InferenceEngineFaceReco* is notified.

InferenceEngineFaceReco

The *InferenceEngineFaceReco* contains the rule set for generating recommendations to add or remove a contact from a face based on the classification and analysis results. The *InferenceEngineMessageAnalysis* produces the input data for this class, which consist of three parameters: the result of the classifier, the amount of spelling errors and the number of typographical signs.

For each user of CoDaMine one instance of this inference engine is launched. Each instance observes events concerning the users' contacts. Each contact has a

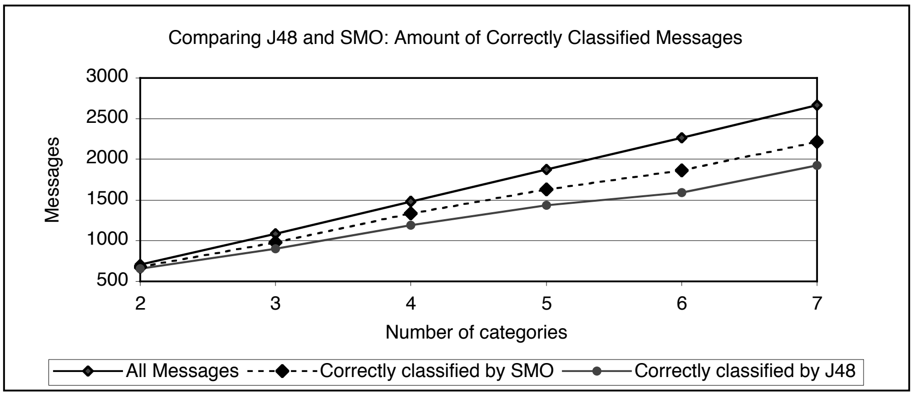

Figure 4. Accuracy comparison of J48 and SMO.

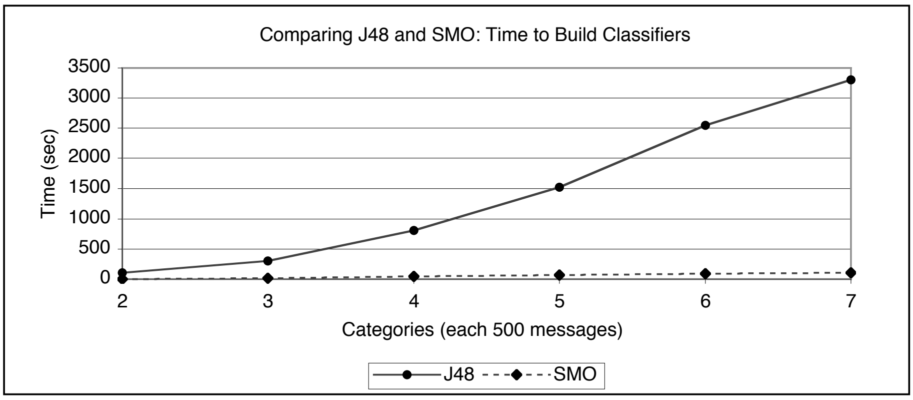

Figure 5. Time cost comparison of J48 and SMO.

unique relation to the user based on the communication data and the assigned faces, which are represented by categories. These relations are mapped to a score system. Each time a result from the message analysis module is received, the contained parameters are used to update the score of its sender. Figure 6 shows the processing of an incoming event in the *InferenceEngineFaceReco*.

The concerning contact is retrieved from a list, in which all contacts are stored as objects. Based on the parameters of the event (i.e., the classification result in the form of the face that the system recommends, the results of the typographical sign and of the spelling error analysis), the overall evaluation of the respective message is determined. The overall evaluation is represented as a number of points. These points are added to the score of the contact. Now the original face of the message is compared with the result of the classification. If the result of the classification does not correlate with the origin face, the score of the origin face gets decreased. This allows—although unlikely—to generate a recommendation to remove a contact from a certain face. The internal thresholds are the following: if a new classified message raises the score above a value of +100 points, the face is recommended; if the score falls below the threshold of -70, removing the sender from the face is recommended. In the end the threshold is compared with the shifted scores. If the threshold is reached, the system generates a sensor event and resets the score just for the case the user rejects a recommendation and the system has to perform the decision process again. This event is retrieved by the client and presented as a popup dialog, if the contact is not already contained in the target face.

If the user rejects a recommendation, a feedback `SensorEvent` is sent from the *RecommendationSensor* to the inference engine. This results in an increased threshold for the contact and the recommended face. This way it is possible that the same recommendation can be made again in the future but later and based on more data.

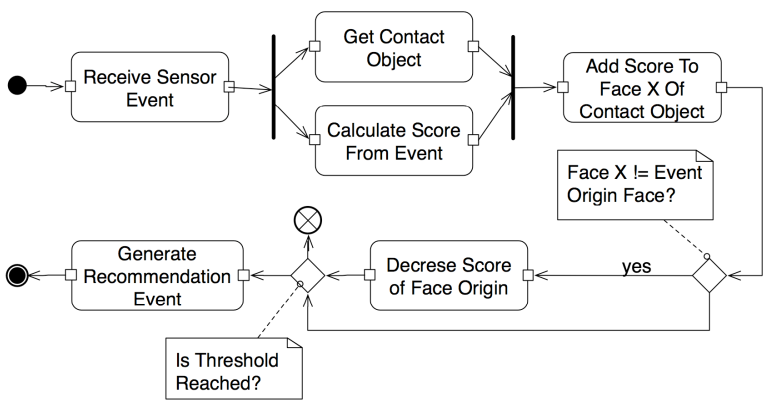

Figure 6. Activity diagram of the InferenceEngineFaceReco.

6 Related Work

Privacy issues as well as trust and security are an on-going focus of ubiquitous computing research. Often addressed only on the behalf of a single, specific system and its consequential privacy concerns [17], lately some more complex concepts and models are introduced. Yet, these models often have a very specific and focused view on the subject or are too ambitious and abstract in order to be realisable in near future.

The trade-off between privacy and mutual awareness is a central question discussed in CSCW research. Sellen and Belotti [3] address this problem and transfer it to the field of collaborative ubiquitous computing environments delivering a design framework for their principles of control and feedback. By formulating eight design question to analyse the control users have on their outgoing information, and the feedback they receive on how and by whom this information is used. Applied to CoDaMine we can state that PRIMIFaces already was obliged to the principles of feedback and control. CoDaMine even advances this effort by enhancing the quality of feedback the user gets.

Founded on the ‘Principle of Minimum Asymmetry’ Jiang et al. [26] describe a model of ‘Approximate Information Flow (AIF)’ designed to reduce the asymmetry in which the members of a ubiquitous computing system are informed about each other. In order to achieve this balance the model includes three abstract views on the information flow in ubiquitous computing architectures addressing where and how data is stored (called information spaces), the lifecycle from collection over access to second use and accordingly the themes for achieving the desired minimum asymmetry. As mentioned before, giving feedback to the users is crucial in order to develop trustworthy systems. But in order to support collaboration between users, there is no need for enforcing symmetry, rather the social processes between users should be considered. CoDaMine supports users in their decisions and gives them final control about which and how much information is disclosed in each context.

The Privacy Awareness System (pawS) described by Langheinrich [18] is a technical concept enabling privacy management based on the wireless exchange of machine-readable privacy policies derived from the W3C Platform for Privacy Preferences (P3P) [5]. Founded on these policies services negotiate the information exchange between users and sensors in the environment. Users have to maintain a set of general and specific rules. The authors do not indicate to what extent users can restrict the access to a certain group of users and to disclose the same information to another group.

In ‘Everyday Privacy in Ubiquitous Computing Environments’ Lederer et al. [20] discuss the idea of bringing Goffman’s theory of faces to the field of ubiquitous computing. Yet overall their approach is limited to three distinct settings (no information, vague information, full information).

7 Conclusions and Future Work

In ubiquitous environments an increasing number of sensors capture information on users and an increasing number of actuators are available to present information to users. With CoDaMine we presented a concept and an implementation of higher-level inference seamlessly embedded in an infrastructure to support light-weight privacy management. It is based on PRIMIFaces and its faces concept, but goes way beyond with its inferencing approach. We showed a way in which inference about social connections between users of ubiquitous computing environments can help to support these users to make well-informed decisions on their privacy. Overall CoDaMine does not aim to directly deal with the operationalisation and measurement of privacy—rather, CoDaMine empowers users to make informed decisions concerning their information disclosure.

The CoDaMine concept has one obvious limitation: it is based on the inferencing on users’ online communication. It, therefore, only works if users actually communicate with each other. The concept cannot capture the evolution of social relationships that occurs non-verbally.

In the future work we aim to integrate new inferencing algorithms and additional sensor types capturing further types of information. For example, CoDaMine considers the number of spelling errors in a message as a form of noise irritating the text mining, and therefore minimising the effectiveness of the classification. As this has some drawbacks at the moment, we are currently looking into possibilities to capitalise on this behaviour by counting the same spelling error and deriving characteristics of a face. If the alleged error keeps reappearing in the same diction, it then should increase the weight of the message instead of lowering it. In this way we could enhance the quality of recommendations by refining the algorithms of CoDaMine. The ability of natural language processing systems—mostly using an integrated lexicon to look up words—to differentiate between spelling errors and unknown words like proper names or abbreviations is currently often limited. For example, the Jazzy library used in CoDaMine is able to handle emoticons and abbreviations, but cannot properly handle names.

Another aim is to integrate the sensors and inference engines we developed in the field of location-awareness, giving information on dependencies between places and faces. By assigning places to faces and inferring the phases of spatial co-location between the contacts, and combining this with the results of CoDaMine, the system could get more insights on the social coherences between the users.

Overall the concept and system are fully developed and implemented and we have tried it out with test data. However, a long-term study is still missing. Such a long-term study is important to reveal the real benefits of using CoDaMine for privacy and trust management in ubiquitous environments.

Acknowledgments

We thank the members of the Cooperative Media Lab; special thanks to Julian Seifert who worked on CoDaMine in his BSc thesis, and to Thilo Paul-Stueve for support with SensBase and comments on earlier versions of the paper. Part of the work reported in this paper has been funded by the EU Socrates Minerva project Adaptive Learning Spaces (ALS) (229714-CP-1-2006-1-NL-Minerva). Thanks for the anonymous reviewers for comments on earlier versions of this paper.

References

1. Apache Software Foundation. *ws-xmlrpc - Apache XML-RPC*. http://ws.apache.org/xmlrpc/, 2003. (Accessed 10/7/2007).
2. Baym, N.K. *The Emergence of On-line Community*. In Jones, S., ed. Cybersociety 2.0: Revisiting Computer-Mediated Communication and Community. Sage Publications, Inc., Thousand Oaks, CA, USA, 1998. pp. 35-68.
3. Bellotti, V. and Sellen, A. Design for Privacy in Ubiquitous Computing Environments. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW'93*. pp. 77-92.
4. Bergs, A. *Analysing Online Communication From a Social Network Point of View: Questions, Problems, Perspectives*. 2006.
5. Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M. and Reagle, J. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. http://www.w3.org/TR/P3P/, 2002. (Accessed 10/7/2007).
6. Friedman, B., Kahn, P.H. and Howe, D.C. Trust Online. *Communications of the ACM 43*, 12 (Dec. 2000). pp. 34-40.
7. Goffman, E. *The Presentation of Self in Everyday Life*. Doubleday Anchor Books, N.Y., 1959.
8. Gross, T., Egla, T. and Marquardt, N. Sens-ation: A Service-Oriented Platform for Developing Sensor-Based Infrastructures. *International Journal of Internet Protocol Technology (IJIPT) 1*, 3 (2006). pp. 159-167.
9. Gross, T. and Oemig, C. PRIMI: An Open Platform for the Rapid and Easy Development of Instant Messaging Infrastructures. In *Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications - SEAA 2005*. pp. 460-467.
10. Gross, T. and Oemig, C. From PRIMI to PRIMIFaces: Technical Concepts for Selective Information Disclosure. In *Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications - SEAA 2006*. pp. 480-487.
11. Herring, S.C. *Computer-Mediated Discourse*. In Tannen, D., Schiffrin, D. and Hamilton, H., eds. Handbook of Discourse Analysis. Blackwell Publishing Limited, Oxford, 2001. pp. 612-634.
12. Hidalgo, J.M.G., Bringas, G.C., Sánz, E.P. and García, F.C. Content Based SMS Spam Filtering. In *Proceedings of the 2006 ACM symposium on Document engineering*. pp. 107 - 114.
13. Idzelis, M. *Jazzy - The Java Open Source Spell Checker*. http://jazzy.sourceforge.net/, 2006. (Accessed 10/7/2007).
14. Ignite Realtime. *Wildfire*. Jive Software Community, http://www.jivesoftware.org/wildfire/, 2006. (Accessed 10/7/2007).
15. Jabber Inc. *Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP*. http://www.jabber.org/, 2007. (Accessed 10/7/2007).
16. Jiang, X., Hong, J. and Landay, J.A. Approximate Information Flows: Socially-Based Modelling of Privacy in Ubiquitous Computing. In *Proceedings of the Fourth International Conference on Ubiquitous Computing - UbiComp 2002*. pp. 176-193.
17. Langheinrich, M. Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. In *Proceedings for the Third International Conference on Ubiquitous Computing - UbiComp 2001*. pp. 273-291.
18. Langheinrich, M. *When Trust Does Not Compute - The Role of Trust in Ubiquitous Computing*. Presented at Workshop on Privacy at the Fiveth International Conference on Ubiquitous Computing - Ubicomp 2003, (Oct. 12, Seattle, WA). Canny, J., Dourish, P., Grossklags, J., Jiang, X. and Mainwaring, S., eds. 2003.
19. Langheinrich, M. *Personal Privacy in Ubiquitous Computing: Tools and System Support*. Ph.D. thesis, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 2005.
20. Lederer, S., Dey, A.K. and Mankoff, J. *Everyday Privacy in Ubiquitous Computing Environments*. Presented at Workshop on Socially-informed Design of Privacy-Enhancing Solutions in Ubiquitous Computing at the Fourth International Conference on Ubiquitous Computing - Ubicomp 2002, (Sept. 29, Göteborg, Sweden). 2002.
21. Paolillo, J.C. The Virtual Speech Community: Social Network and Language Variation on IRC. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences - HICSS'99*. pp. 2028.
22. Patrick, P.L. *The Speech Community*. In Chambers, J.K., Trudgill, P. and Schilling-Estes, N., eds. The Handbook of Language Variation and Change. Blackwell Publishing Limited, Oxford, 2002.
23. Platt, J.C. *Fast Training of Support Vector Machines Using Sequential Minimal Optimisation*. In Schoelkopf, B., Burges, C. and Smola, A., eds. Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA, 1999. pp. 185-208.
24. Shin, D.-S. ESL Students' Computer-Mediated Communication Practices: Context Configuration. *Language Learning & Technology 10*, 3 (Sept. 2006). pp. 65-84.
25. Tosca, S. The EverQuest Speech Community. In *Proceedings of the Computer Games and Digital Cultures Conference*. pp. 341-354.
26. Voida, A., Newstetter, W.C. and Mynatt, E.D. When Conventions Collide: The Tensions of Instant Messaging Attributed. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2002*. pp. 187-194.
27. Witten, I.H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2006.