Chapter 23
Context and Community Awareness in Support of User Intent Prediction

Nikos Kalatzis, Ioanna Roussaki, Nicolas Liampotis, Pavlos Kosmides, Ioannis Papaioannou and Miltiades Anagnostou

Abstract Proactive behaviour of pervasive computing systems cannot be realised without the establishment of suitable and reliable user intent prediction facilities. Most of the existing approaches focus on an individual end-user's history of interactions and context in order to estimate future user behaviour. Recent trends in pervasive systems allow users to form communities with other individuals that share similar profiles, habits, and behaviours. Pervasive Communities set new challenges and opportunities regarding proactivity and context management. This chapter presents a context aware user intent learning and prediction framework that is able to exploit the knowledge available at the community level. Community knowledge, if appropriately managed, can significantly improve proactivity behaviour of individual users' systems.

23.1 Introduction

Context awareness, combined with learning and inference mechanisms, contributes greatly to establishing the proactive behaviour of pervasive systems, thus minimising the necessary human-machine interactions and providing an improved user experience. Various research outcomes indicate that repeated patterns can be usually

I. Roussaki (✉) · N. Kalatzis · N. Liampotis · P. Kosmides · I. Papaioannou · M. Anagnostou
National Technical University of Athens, Athens, Greece
e-mail: ioanna.roussaki@cn.ntua.gr

N. Kalatzis
e-mail: nikosk@cn.ntua.gr

N. Liampotis
e-mail: nicolas.liampotis@cn.ntua.gr

P. Kosmides
e-mail: pkosmidis@cn.ntua.gr

I. Papaioannou
e-mail: jpapai@cn.ntua.gr

M. Anagnostou
e-mail: miltos@cn.ntua.gr

P. Brézillon, A. J. Gonzalez (eds.), *Context in Computing*,
DOI 10.1007/978-1-4939-1887-4_23

detected in human behaviour (Magnusson 2004). People follow their own routines of actions to perform everyday tasks. Modelling, recording and applying learning algorithms on datasets representing user interaction histories along with the corresponding context information can lead to the discovery of user behaviour models.

Most current pervasive computing systems have been designed mainly to address the requirements of individual users. This approach, however, does not consider the need of humans to communicate and socialise with others. To this end, the notion of pervasive communities (Doolin et al. 2012) is introduced that aims to bring together pervasive computing systems and social media, in order to support various interactions among users, communities, resources and smart environments. This notion is modelled based on the Cooperating Smart Space (CSS) and the Community Interaction Space (CIS) paradigms that have been introduced (Doolin et al. 2012). Each individual user is represented by a CSS that may consist of several nodes (devices). Each CIS on the other hand corresponds to exactly one pervasive community and may contain multiple CSSs. Individual users (CSSs) may be part of various communities (CISs) simultaneously. CSSs and CISs aim to support the formation of dynamic physical or virtual pervasive communities of users that demonstrate commonalities for a non-trivial period. The formation of communities can be based on various criteria, which are context-related in most cases (e.g. location, interests, preferences, beliefs, ideas, goals, experiences, etc.) Thus efficient context representation and management on a user and community level is one of the most important features of CSSs/CISs in properly adapting the services provided to users, but also in orchestrating the pervasive communities (Roussaki et al. 2012).

In order to support the proactivity and personalisation-related facilities of pervasive communities, a suitable user intent model is necessary. Thus, the Context Aware User Intent (CAUI) data model has been introduced to capture the common behaviour patterns that may exist for a single user or for a community of users. The CAUI data model primarily describes the actions that a user performs and the possible sequences across those actions along with the accompanied context.

Currently, predicting the future behaviour of humans in pervasive systems has been limited to individual user level. Very few research initiatives have dealt with applying such techniques to communities of end-users in order to extract collective behaviour models regarding interactions with pervasive computing services. This introduces several research challenges, as it needs to take into account social aspects, user similarities and more personal goals, thus having the potential to support both communities and individuals in multiple ways. Knowing the typical behaviour of a community member allows new members to benefit from the existing community's experience. The community behaviour model can be used as an intermediate measure for the time period in which more accurate data are gathered and until a user behaviour model has been generated. Therefore, long learning periods are avoided and predictions are feasible even in situations that the user has not been in the past. On the other hand, serious shortcomings are arising as it is not always safe to predict user behaviour based on community knowledge. It is necessary to develop decision mechanisms that indicate whether community originated knowledge is useful.

This chapter researches the issue of community and context awareness in sup-port of user intent prediction for users interacting in pervasive computing environments. More specifically, the chapter is structured as follows. Initially, a state of the art review is presented. Then, an illustrative use-case scenario is described and the most important user and technical requirements are extracted. The next section elaborates on user and context modelling and the context-based user/community intent learning and prediction mechanisms that have been established are presented. The evaluation of the proposed mechanisms is presented in the next section and finally, the chapter's conclusions are drawn and the respective future plans are discussed.

23.2 Related Work

Various successful paradigms exist where community knowledge is utilised in support of the individual. Authors in Thakor et al. (2004) examined the interaction of users with simple software systems (e.g. web browsers) and concluded that user characteristics affect the way of interaction. Thus, users demonstrating profile similarities tend to interact in a similar manner for achieving certain tasks allowing the proactive adaptation of services according to user characteristics. Collaborative recommender systems, known as Collaborative Filtering (CF) (Adomavicius and Tuzhiin 2005), are another successful paradigm of community knowledge extraction and exploitation. These systems estimate users' unknown ratings over items based on known ratings of similar users.

Pervasive computing systems understand and describe user intent as the tool employed to assist a user to perform a certain task or achieve a specific goal (Sousa et al. 2006). Several areas of pervasive computing, where user intent has been presented include but are not limited to Smart Home environments (such as MavHome; Gopalratnam and Cook 2007 and Aware Home; Abowd et al. 2002), elderly assistive living (Ni et al. 2011), smart applications (Garlan et al. 2002) etc. To this purpose, user intent is subject to advanced prediction methods, as on deciding on the "next step(s)" of a user based on his/her preferences and the surrounding context information, in an effort to maximise the performance of a system and the overall user experience.

There have been some research efforts that utilise multiple users' interaction histories in support of individual user. Intention prediction of user interactions with information systems is the purpose of the research work presented in (Antwarg et al. 2012). Several personal static profile attributes, like user's age, gender, and other demographic data are incorporated in the model, as the user intent accuracy is highly affected by his/her preferences. As the authors indicate, a user's attributes and context (such as age or the operating system) indicate which sequence of actions the user will eventually perform. The authors model the unique characteristics of the end user in relation to the sequence of user actions with the information systems in order to provide accurate predictions. The mechanism employed is based on hidden Markov models (HMM) on the sequence of observations of the actions that the end-user does, as the sequence of actions is not always observable. The model ends up with attribute-driven HMM trees for intention prediction, while the authors are planning to overcome the limitation of their algorithm with regards to the multitasking capabilities of most concurrent devices and applications. Another user intent prediction approach that resembles common sense reasoning is presented in the LifeNet project (Singh and Williams 2003). This large scale system collected common sense actions/situations from human users, represented the knowledge gathered as a graphical model, allowing the execution of user intent predictions for the short future and only for statistical methods. For example if a user declares she is thirsty then she should possibly seek something to drink. Thus, depending on the user's location, the system can predict the user intent and propose the most convenient action. Similarly, another large dataset has been presented in Eagle et al. (2008), where physical proximity and emotions were combined with 3,30,000 hours of continuously recorded mobile phone data on the actions of real users, self-reports and plain logs. These data along with the recorded behaviour of the user can be employed as pattern for user intent prediction, although much work has to be done in order to be easily maintainable and usable from real applications. In Tang and Liu (2009) the authors collect data to allow for online behaviour prediction of users in social media. The "training" dataset is the information provided by other actors with similar characteristics and preferences, along with affiliations of these sample actors with the subject users. The k-means variant algorithm is employed to handle scalability issues, as the social network is a connected graph with areas of various densities with regards to affiliations of users. The approach is promising but further improvements should be performed in order to provide reliable and rapid user intent predictions.

23.3 Scenario and Requirements

In this chapter a use case scenario is presented based on which specific architecture requirements will be extracted regarding the user behaviour modelling and system's proactivity functionality.

23.3.1 Scenario

Scene 1 The main actor of the scenario, Tom, is living in a "smart home" environment and interacts with various intelligent devices, services, sensors, actuators etc. Tom is currently working on a desktop PC and is about to leave home. He sets the

PC in sleep mode, turns off the radio, turns off room lights and is heading out of the house.

Scene 2 As he approaches his car, he unlocks the door, enters the driver seat, and starts the engine, the navigator, and the radio.

Scene 3 Tom aims to drive at **his** workplace but the navigation system notifies him that a significant number of other drivers with similar profiles (similar origin-destination, similar vehicle type, driving profiles) that are currently in the same area are following an alternative route than the usual. Tom is checking social media messages posted by people in the area and he finds out that the last two days construction works are causing delays. Tom agrees to follow a set of actions followed by other users including the alternate route selection, driving related actions regarding route parts that demand special attention, etc.

Scene 4 Tom makes a stop at the gas station, alarm lights are turned on and mutes the car radio.

Scene 5 When he arrives at work, he turns off all car systems following the usual sequence of actions.

Scene 6 During weekends he usually takes the bus in order to visit a nearby park. While he is at the bus station he checks the time table, purchases an electronic ticket for the short trip and validates it upon entering the bus via his smart phone.

The described interactions, along with the respective context data, are recorded on a personal history log which is maintained on the user's CSS. The Proactivity component provided by the CSS platform exploits recorded data in order to discover Tom's interaction model and provide services with dynamic behaviour features. According to the frequency of occurrences for each action or group of actions, respective probabilities are calculated and the prevailing patterns are included in the interaction model. A graphical representation that reflects the actions described in the scenario is presented in Fig. 23.1. Six groups of actions have been illustrated, each consisting of a task, while user friendly names have been assigned. Task formation is based on context criteria (e.g. time, location) and sequence of actions frequency of occurrence.

It is obvious that certain actions described in the scenario are context depended while others are not. As it is illustrated in Fig. 23.2 the probability of occurrence for some actions contained in the task labelled as "Leaving home", in the first scenario scene, is differentiating according to current context, such as human presence. In case another person is still in the room, the transition probabilities to action labelled as "Turn off lights" are zero.

Scenario scenes 2 and 5 describe driving related sequences of actions that can be similar over a wide spectrum of drivers and context. Starting or stopping a car includes a set of actions that are more or less the same regardless of the context. On the other hand, in scene 6, the task is mainly context based but does not depend on the sequence of previously performed user actions or tasks. Whenever the user stands in a bus station it is highly probable that he will check the bus time table and purchase a bus ticket.

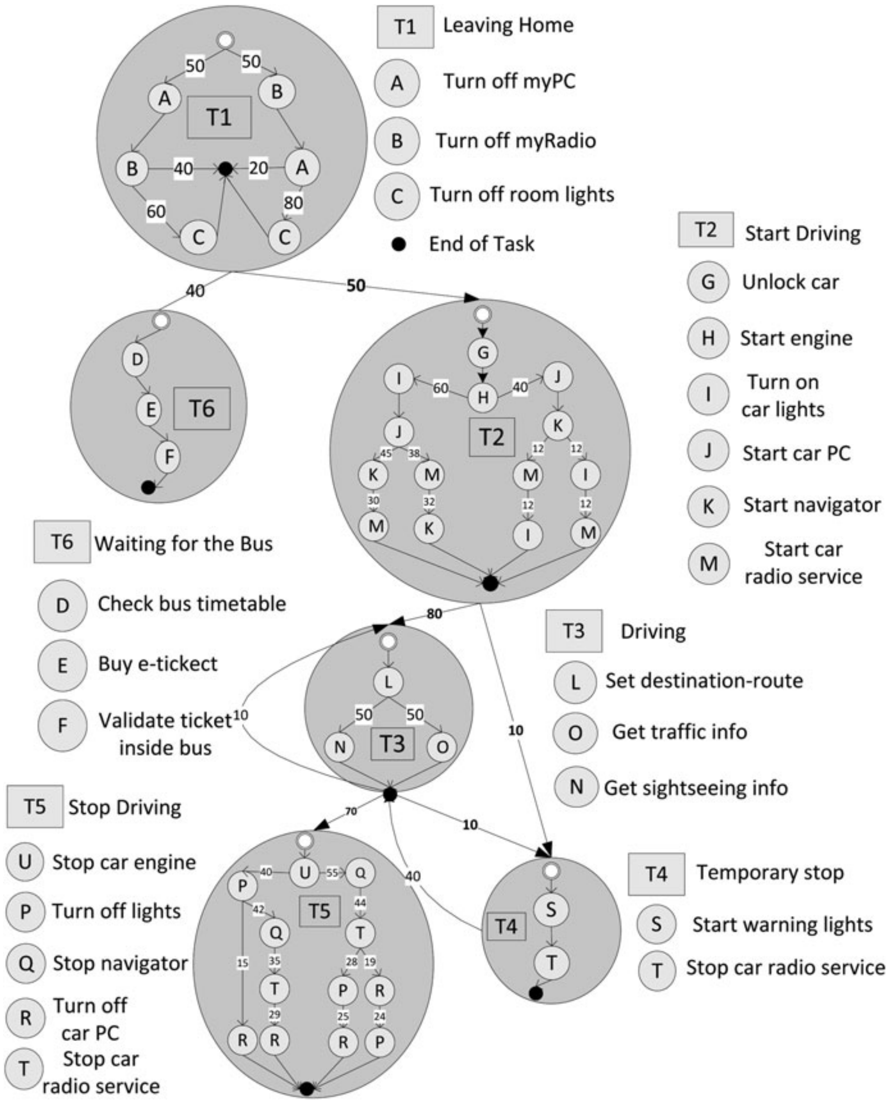

Fig. 23.1 A graphical representation of a single user's actions routines according to the described scenario

In scene 3, Tom is receiving recommendations to follow a set of actions that he does not usually perform. This sequence of actions has been constructed based on routines followed by a significant number of other users with similar profiles that have been on the same situation in the recent past.

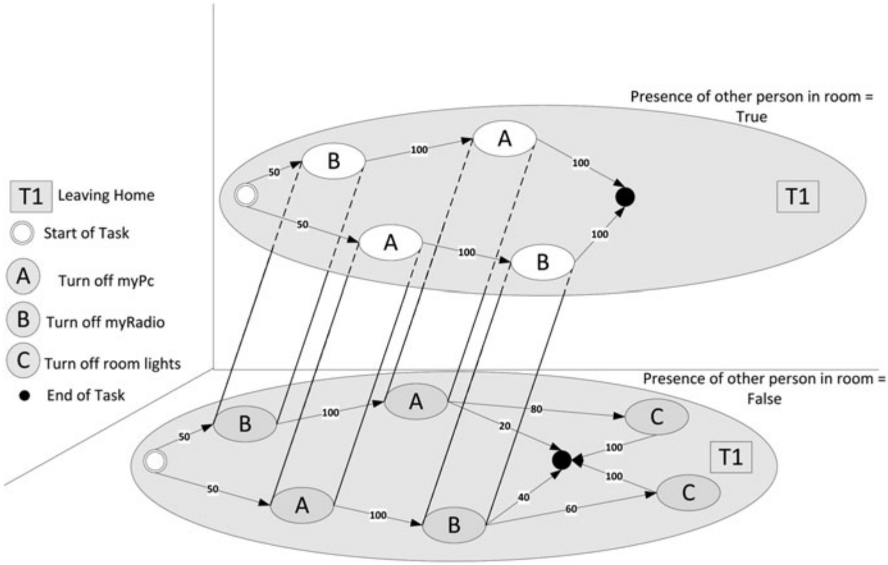

Fig. 23.2 Task "Leaving Home" is differentiated based on context (in this example context regarding the presence of other person in room is considered in the action prediction process)

23.3.2 Requirements

Based on the described scenario, the most important requirements that a proactive system should satisfy are specified hereafter:

User Interaction Modelling and Management User actions and context should be modelled in a manner that will allow the efficient management of data (e.g. to support creation, retrieval, update and deletion operations), to accommodate prediction mechanisms and be easily comprehensible by users.

Discovery of User Intent Model (including community knowledge) Learning algorithms should be able to process recorded data sets of interactions in order to extract user intent models. Data mining techniques and other statistical methods may be applied in order to discover often occurring patterns of actions and context. Classification algorithms, association rules mining, clustering techniques are some of the most common methods used in current state of the art approaches. In a similar manner, the learning algorithm should be able to process history logs or existing prediction models derived from various users in order to discover community-wide recurring patterns of interactions.

Predicting User Actions Discovered interaction models are utilised in order to provide predictions. Predictions should be accompanied by a metric representing the confidence level which will be used by services or humans in order to take further actions. Behaviours of humans may demonstrate significant variations. A person's routines may change over time or even not exist at all, in a significant and tractable level (Horvitz et al. 2002). Hence, it is not feasible to expect that a user interaction predictor will present high success rates in all kinds of situations. As ubiquitous computing systems aim to support persons in everyday tasks, inaccurate prediction should be carefully handled. In a similar manner, community knowledge is not always eligible in user intent prediction and thus should be carefully utilised.

User Control The user should be able to observe learned interaction models but also recorded data sets containing histories of actions. Control over the proactivity system increases user's acquaintance and hence reassures further system's utilisation (Gallacher et al. 2011). It is also important that the user can enable and disable the overall monitoring and prediction system. Community interaction models are built on various users' interaction data. Thus, privacy protection mechanisms should apply. In addition, user should be able to revise (e.g. add, modify, remove) prediction rules. User defined rules, encompassing the notion of goal knowledge, greatly enhance and at the same time simplify models of action.

User feedback Prediction components should be able to recognise and utilise possible successful or unsuccessful predictions. This can be achieved in an automated (implicit) way by monitoring the performed action and comparing it with the predicted or based on explicit user feedback. The latter demands user interference and can be highly distractive if it is not properly handled. Recorded feedback is a useful source of information for improving future learning procedures.

Smoothing on Routines Change Previously learned patterns may no longer reflect the current user interaction routines. People often change their life patterns under certain circumstances (holidays, job change, moving to another place, etc.). The system should be able to detect and adapt based on this changes in a seamless fashion. Furthermore, it should be able to detect and handle appropriately extreme or unexpected recorded values.

Context Selection It is important to select the appropriate accompanying context types in order to construct situation snapshots. It is common to precede learning with an attribute selection stage that aims to eliminate all but the most relevant attributes of the training dataset. As stated in Witten et al. (2011), the best way to select relevant attributes is manually, based on a deep understanding of the learning problem and what the attributes actually mean. As this in not always efficient various statistical and data mining methods have been developed.

23.4 Behaviour Modelling

The introduced Context Aware User and Intent (CAUI) model aims to describe actions that a user performs and possible sequences that arise among those actions. The often occurring sequences of actions are modelled as tasks. The previous sections presented various examples and requirements regarding actions and tasks. The CAUI model is generic enough to model any kind of user action. However, the current framework aims to model and support the prediction of actions that are related with the use of services provided by the CSS platform or third parties. These actions have previously been monitored by software or hardware agents that are part of the CSS platform. A description of the most important data classes of the CAUI model follows:

IUserIntentAction This interface and the respective realisation class models the action that a user performs. The data object contains, among others, information regarding the actual action, the targeted service, user friendly names and unique identifiers.

IUserIntentTask This interface and the respective realisation class models a set of actions (each modelled as IUserIntentAction) that a user performs in order to achieve a certain Task. Additional information captured by this class is related with context (including time) information, user friendly names reflecting goal knowledge, targeted services, and task identifiers.

ITransitionData This interface and the respective realisation class contain all necessary information in order to associate actions or tasks. It refers to a unique transition among actions or tasks. Each instance of this class is related to one source (i.e. a single action or task or groups of these) and exactly one destination.

ICommIntentAction This interface and the respective realisation class model a community action. The CommIntentAction class extends the UserIntentAction class with attributes describing the level of commonality of the performed action among the community members.

ICommIntentTask In a similar manner, the CommIntentTask extends the UserIntentTask with additional attributes referring to the commonality of task.

The UML diagram depicted in Fig. 23.3 illustrates the described Context-Aware User Intent data class model along with the community related extensions.

23.5 Architecture

In this section the main functional parts of the Proactivity component will be presented, as depicted in Fig. 23.4. The provided functionality includes discovery, management and evaluation of user intent models and eventually estimation of the actions that need to be taken. The Proactivity component interacts with the Context Management (CM) component and with potential third party (3P) services that adapt their behaviour according to provided predictions. User interactions with 3P services are monitored, modelled and stored in a history data set by the CM system. The data set is then used for discovering the CAUI model. Finally, a GUI allows users to directly interact with the underlying mechanisms.

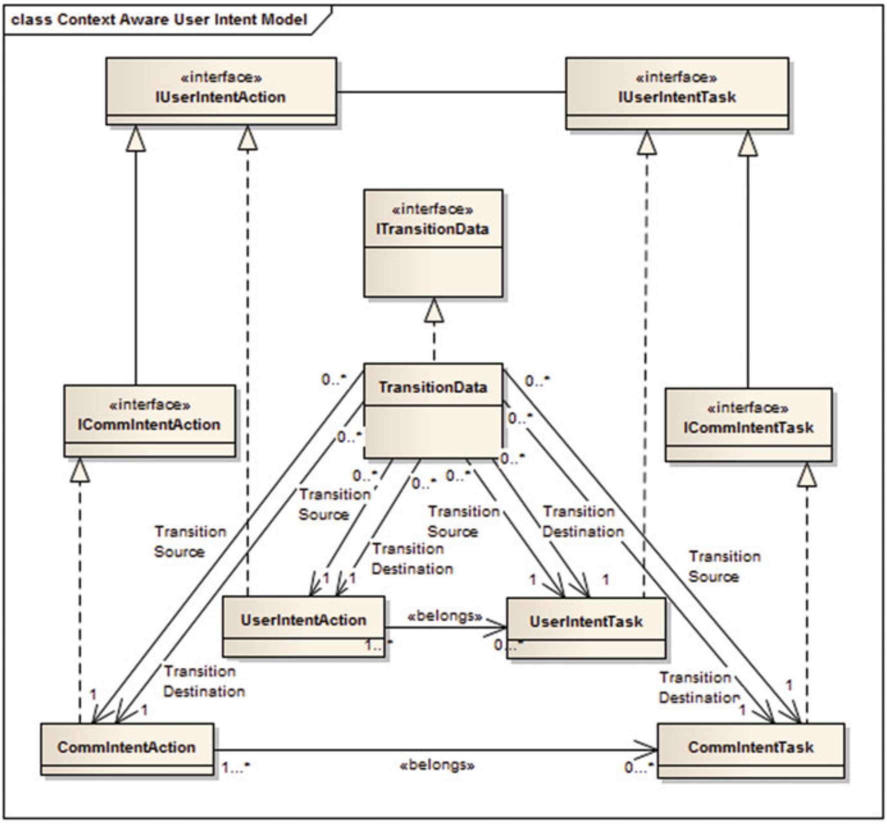

Fig. 23.3 Context-Aware User Intent (CAUI) Data class model

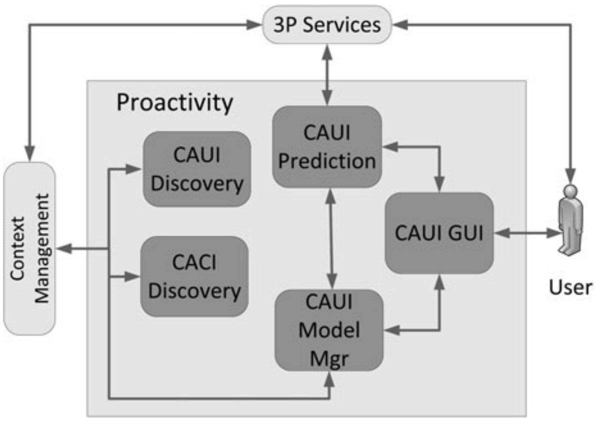

Fig. 23.4 Functional architecture of the Proactivity component block

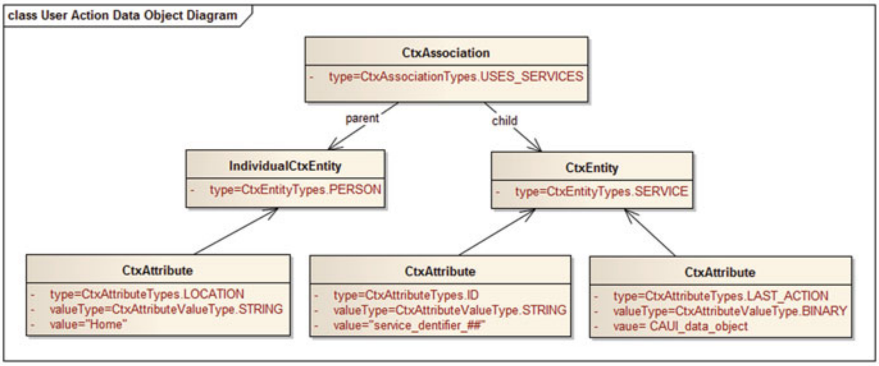

Fig. 23.5 Context model usage for user and services interactions

23.5.1 Context Management and Modelling

The CM component provides features for modelling and managing context information within the CSS framework. The context model, detailed in Kalatzis et al. (2014), includes all the necessary concepts in order to efficiently represent, retrieve, exchange, maintain and manage context information in a CSS environment. The context model comprises the following core informational classes: CtxEntity, CtxAttribute, CtxAssociation and CtxHistoricAttribute. Additionally to these classes there is the CtxIdentifier class hierarchy, which enables the identification of context data items in a privacy-aware manner and the CtxQuality class, which further augments the model with Quality of Context properties. The CtxEntity is the core concept upon which the context model is built and corresponds to an object of the physical or conceptual world. Essentially, the CtxAttribute class identifies an entity's status in terms of its static and dynamic properties and it therefore captures all context information items that are used to characterise the situation of the owner CtxEntity.

Persons and communities are modelled based on IndividualCtxEnity and CommynityCtxEntity which are extending the CtxEntity class. To address the requirements regarding context semantics, a taxonomy has been introduced that includes the various context types as tags and dictates how these can be combined (Roussaki et al. 2012). Figure 23.5 illustrates an example of the context model utilisation for modelling user context and interactions with services. In the current example, user actions modelled as CAUI data objects are stored as a CtxAttribute of type "LAST_ACTION" while additional context (e.g. user location) is stored as a CtxAttribute assigned to IndividualCtxEntity of type "PERSON".

Regarding the CM architecture a detailed description can be found in Roussaki et al. (2012). The provided functionality includes the management of current, historic and future context data in a synchronous and/or asynchronous way; inference of high-level context information based on raw sensor data and/or context history; management of context sources and collection of monitored data. In addition, the

CM architecture comprises a Database Management System which enables access to the actual context repositories, i.e. the Context Database and the Context History Database. Both database schemata conform to the context model described in Kalatzis et al. (2014).

23.5.2 CAUI Discovery

The CAUI Discovery module provides the learning mechanisms for constructing the CAUI model. The training data set contains recorded interactions accompanied by a snapshot of context data reflecting the situation (location, time, device used, service type, user status, ambient parameters, etc.) in which the action was performed. The number of data types included in the snapshot, influences the learning process in terms of time and computational resources consumption but also in terms of the prediction model completeness, thus a balance must be maintained.

Representing high-level context information based on captured low-level data may include uncertainty. However, state-of-the-art sensing technologies and inference mechanism allow the accurate estimation for some context types. For example, location and time, which are significantly related with users' actions, can be monitored with high accuracy. In addition, there are several solutions for dynamic identification and assignment of user friendly tags for places of interest. Regarding time, recorded timestamps allow the extraction of additional information such as hour of day, day of week, month, season, weekends, etc.

The learning algorithm should be able address the requirements above and to discover both recurring sequences of actions but also actions that occur in certain situations. We assume that a user interacts with various resources in a pervasive computing environment and his/her interactions are monitored for time $T = t_k$. Let a^{t_k} express the user interaction at time t_k ($k = 1,2, ..., K$), which was observed when the user was in situation $s^{t_k} = \left\{c_1{}^{t_k}, c_2{}^{t_k}, ..., c_n{}^{t_k}\right\}$ where $c_i{}^{t_k}$ is the value of context information of type i, ($i = 1,2, ..., n$). Given all the observed/recorded tuples $\left(a^{t_k}, s^{t_k}\right)$for $k = 1,2, ..., K$, the CAUI Discovery component aims to build a behaviour model to identify the most probable interaction $a^{t_{k+1}}$ that the user will perform. To this end, the learning algorithm processes history logs to extract user action patterns. In the proposed approach, the following cases are targeted: (i) user performs actions in forms of sequences, (ii) user performs actions when certain situations occur, (iii) a combination of (i) and (ii).

23.5.2.1 Sequences Discovery

Based on the described requirements, discrete time Variable Order Markov Chains (VOMC) (Begleiter et al. 2004) has been selected as an appropriate algorithm for modelling and predicting sequences of actions. Prediction of future states of a Markov model depends only on current state, while predictions of a VOMC model depend also on the history of a number of states defined by the model's order. This algorithm is applicable as the system is fully observable and actions are represented as a finite alphabet of possible states $A : \left\{a^{t_1}, a^{t_2}, ..., a^{t_n}\right\}$. Discrete time is also supported as data derived from software or hardware sources/sensors and hence distinct updates of context data are occurring. During the learning phase, conditional transition probabilities for all symbols $a^{t_k} \in A$ are estimated by counting the number of action occurrences appearing after a sequence of actions $d \in A$where $[d : \left\{a^{t_{k-1}}, a^{t_{k-2}}, ..., a^{t_{k-1}}\right\}$ and l denotes the length of d. In the proposed framework, the maximum length of dis specified based on a predefined significance threshold that is calculated taking into account the transition probability value and the actual number of recorded actions occurred. The learned sequence predictor maintains several Markov models grouped based on their order.

23.5.2.2 Situations Discovery

The aim of this process is to identify recurring patterns where specific context value combinations are accompanying a certain action. It is possible to handle this issue as a classification problem, where input instances are the observed context values and the outcome is the category of action. According to Witten et al. (2011), Naive Bayes is an algorithm that fulfils the requirements set earlier and has worked quite well in many complex real world situations. It follows a supervised learning approach for estimating parameters of the classifier, such as means and variances of the variables. The algorithm requires a small amount of training data and provides quantifiable probability distributions for each possible class. In addition, it handles well missing values and automatically ignores irrelevant attributes in a process that resembles attribute selection. Finally there is no need for domain expert interference in designing dependencies between input attributes, something necessary for Bayesian Networks. On the other hand, it assumes that attributes are independent from each other with respect to the classification outcome, something that it is not always the case, while the computing resources consumption can get significantly high. Bayes' rule for calculating prediction probabilities according to the defined problem becomes:

$$P\left[a^{t_k}|s^{t_k}\right] = P\left(a^{t_k}\right) \times \frac{\prod_{j=1}^{n} P\left(c_j^{t_k}|a^{t_k}\right)}{P\left(s^{t_k}\right)} \tag{23.1}$$

where a^{t_k} is the expected classification outcome and $s^{t_k} = \left\{{c_j}^{t_k}\right\}$, $j = 1..n$ is the current evidence input.

23.5.3 *CACI Discovery*

This module aims to create a context-aware community intent (CACI) model by combining common interaction patterns and situations among a group of users. The learned model can then be made available to individual users in order to improve accuracy of user interaction predictions. History logs containing user interactions along with accompanying context are collected to a common repository. To this end, data are pre-processed before being fed into the learning algorithms, in order to apply community-wide semantics. As the community dataset is expected to contain contradicting, incomplete or extreme data, a cleaning phase is necessary in order to drop those tuples. The same learning algorithms used for user intent model discovery are applied for sequences of actions extraction and situations identification.

23.5.4 CAUI Prediction

This module exploits learned models in order to perform next action predictions. Each prediction is accompanied by a confidence level that acts as an indicator for further handling of prediction by users or services. Utilising different prediction methods can be particular useful in cases where one of the predictors fails to provide predictions with high confidence, or in cases where the system has not detected a performed action or any context input. The following main categories of prediction approaches are identified according to available input:

User-Performed action(s) In this case, system is requested to provide a prediction of next user actions based on one or more performed actions. According to the number of actions, the appropriate Markov model is selected and probabilities are extracted. Given a sequence of actions $d : \left\{a^{t_{k-1}}, a^{t_{k-2}}, ..., a^{t_{k-l}}\right\}$ where l is the sequence length, the most probable next action is identified by maximizing the respective probability:

$$\mathrm{a}_{seq}^{t_k} = \arg \max_{a^{t_k} \in A} \{P[a^{t_k}|d]\}$$

$$a_{seq}^{t_k} = \arg \max_{\mathrm{a}^{t_k} \in A} \left\{\frac{N(a^{t_k} \ldots a^{t_{k-l}})}{N(a^{t_{k-1}} \ldots a^{t_{k-l}})}\right\} \tag{23.2}$$

Situation Update In this case, a situation is monitored and the respective context action values are used as instances of the Naive Bayes classification model. Actions demonstrating the highest prediction probabilities are returned. For a situation snapshot described by $s^{t_k} = \left\{{c_j}^{t_k}\right\}, j = 1..n$ and given the Eq. (23.1) the optimal action category is requested based on the respective probability maximization:

$$\mathrm{a}_{sit}^{t_k} = \arg \max_{a^{t_k} \in A} \left\{P\left[a^{t_k}|s^{t_k}\right]\right\}$$

$$\mathrm{a}_{sit}^{t_k} = \arg \max_{a^{t_k} \in A} \left\{P(a^{t_k}) \times \frac{\prod_{j=1}^{n} P(c_j^{t_k}|a^{t_k})}{P(s^{t_k})}\right\}$$

For every action $a_{\text{sit}}^{t_k} \in A$ the value of $P\left(s^{t_k}\right)$ is a constant, so it can be omitted and the equation becomes:

$$\mathrm{a}_{sit}^{t_k} = \arg\max_{a^{t_k} \in A} \left\{ P(a^{t_k}) \times \prod_{j=1}^{n} P(c_j^{t_k} | a^{t_k}) \right\} \tag{23.3}$$

User-Performed Action and Situation Input In this case, both methods are combined. The adopted approach weights each outcome based on the number of occurrences contributed in probability calculation. Given (23.2) and (23.3) the action predicted based on both inputs is estimated by:

$$\alpha_{comb}^{t_k} = \arg\max_{a^{t_k} \in A} \left\{ w_{\text{sit}} \cdot P\left[a^{t_k} | s^{t_k}\right] + w_{\text{seq}} \cdot P[a^{t_k} | d] \right\} \tag{23.4}$$

The amount of history records used for the situation based action prediction is reflected in:

$$w_{sit}(a^{t_k}, s^{t_k}) = \frac{N_{sit}^{a_j^{t_k}}}{N_{seq} + N_{sit}}$$

$$N_{sit}^{a_j^{t_k}} = \frac{\sum_i N_{c_{il}}^{a_j^{t_k}}}{\sum_i N_{c_{il}}} \cdot N_{sit} \text{ and } N_{sit} = [N - N_{\bigcup \overline{c_{il}^{t_k}}}]$$

where $\sum_i N_{c_{il}}^{a_j^{t_k}}$ is the sum of recorded context instances c_{il} occurred along with the predicted action and $\sum_i N_{c_{il}}$ is the sum of all recorded context instances c_{il} regardless of the action. Finally, N_{sit} is the number of history tuples $\left(a^{t_k}, s^{t_k}\right)$ where at least one context value $c_{il}$of current situation is contained in vector s^{t_k} and hence participated in prediction process. The amount of history records used for the particular action prediction based on a sequence of performed actions is reflected in:

$$w_{seq} = \frac{N(a^{t_k} \ldots a^{t_{k-l}})}{N\,(a^{t_{k-1}} \ldots a^{t_{k-l}}) + N_{sit}}$$

where $N_{seq}^{a_j^{t_k}}$ is the number of sequences containing the action a^{t_k} and N_{seq} is the number of sequences that the user has currently performed.

Community Assisted It is possible that all described methods do not provide adequate results or that a prediction model does not even exist. In this case, community knowledge can be similarly exploited and the provided results can be compared and combined in order to improve future action estimation.

23.5.5 CAUI Model Manager

This module provides the necessary functionality for instantiating and managing CAUI models. It provides methods for creating, retrieving, updating, and removing data objects referring to CAUI model classes. Retrieval of model objects is performed based on various criteria such as IUserIntentAction or IUserIntentTask identifiers, user action details, service type, maximum probability of occurrence etc. This module is utilised by CAUI Discovery component during model generation or update in order to construct the structure of actions, tasks and the respective transitions among them, or to remove an obsolete model. CAUI Prediction mainly utilises retrieval methods in order to identify actions and tasks that meet specified criteria.

23.5.6 CAUI User Interface

The scope of this module is to enhance the user's trust in the overall framework by making it transparent and easy to control. As already stated, user should be able to know what the system has learned, why it proceeds to certain decisions and to enable and disable it at any given time. To this end, CAUI GUI allows user to control various aspects of Proactivity components functionality. The GUI is implemented as a web interface that visualises CAUI model aspects and provides options for manipulating actions and tasks. The GUI supports prediction rules creation and also prompts for user feedback when prediction confidence level is low. It also allows the enabling and disabling of user monitoring, future action prediction functionality and the discovery of new intent models. Finally, it provides a log of performed and predicted actions along with accompanied context.

23.6 Evaluation

The proposed user intent prediction mechanism has been evaluated based on a data set originally collected for the needs of the Reality Mining project (Eagle et al. 2008) of the MIT Media Laboratory. The data set includes data collected by 94 individuals that were using mobile phones with pre-installed software capable of recording various context attributes such as location, voice and data calls, mobile phone application usage, etc. Their activities and interactions with their mobile devices have been monitored for a 10 month period (i.e., from September 2004 until June 2005) and have been recorded. Additional information such as friendship and proximity among subjects has been recorded as well, also based on questions answered by participants. The datasets of these individuals contain tuples of the following context attributes: ApplicationID, Day of Month (DoM), Hour of Day (HoD), location cell ID. To this extent, it has been decided to use the data collected in the first weeks of the 10 month user monitoring period for training purposes that led to a training data set of about 5,00,000 tuples and use the remaining 10 % for evaluation.

Three sets of experiments have been conducted for evaluating user action prediction. Initially, only the sequence of previously performed user actions was used as input. Then, situation-based predictions were tested, exploiting current context data (i.e. location, HoD, and DoW). Finally, the proposed hybrid approach was tested coupling sequence- and situation-based results. For the second set of experiments, an iterative process has been adopted, where users' training data have been collected in a common data set and fed to the three learning algorithms. In Kalatzis et al. (2014), the authors utilised a heuristic algorithm to identify groups of users demonstrating common characteristics in terms of context and friendship connections. For the needs of this paper, the user group carrying the strongest similarity among its members has been selected. This user group contains 22 individuals. The discovered prediction models were evaluated using the same evaluation data set as in the first experiments. The common data set repository did not contain the training data set of the user under evaluation, thus resembling a community assisted prediction process for situations that the user has never encountered.

The results of both experiment sets are illustrated in Fig. 23.6, where the average prediction success rations (*left axis*) and prediction attempt ratios (*right axis*) for the three strategies are provided. In both the community-unaware, as well as the community assisted approaches (presented in Fig. 23.6a, b respectively), the hybrid mechanism clearly outperforms the other two strategies regarding the percentage of attempted predictions, as it delivers 18–22 % more predictions in (a) and 11–97 % more predictions in (b) in average. On the other hand, regarding the prediction success ratios, the sequence-based mechanism slightly outperforms the hybrid mechanism, as it achieves about 4 % higher success rate in average for (a) and 3 % for (b). However, comparing these two dominant strategies, as the success rates achieved are comparable, we may conclude that the hybrid mechanism is preferable, in case the user values greatly the proactive behaviour of the system. Regarding the context-based predictions, one can easily observe that the achieved results are always outperformed by the other two strategies. This is mainly due to the nature of the data set, where the user actions were minimally depending on the user situation, thus forcing our mechanism to build Bayesian graphs of numerous edges with very low probabilities. Finally, it should be highlighted that the proposed approach delivers quite satisfactory results in the community-assisted case (Fig. 23.6b), managing to correctly predict the user actions in two out of three cases, attempting to make predictions in about one out of two cases in average, using the hybrid mechanism. This is of high value, given the fact that zero user historic data are assumed, and all user action predictions are performed based on the user intent models built for the specific user's fellow community members.

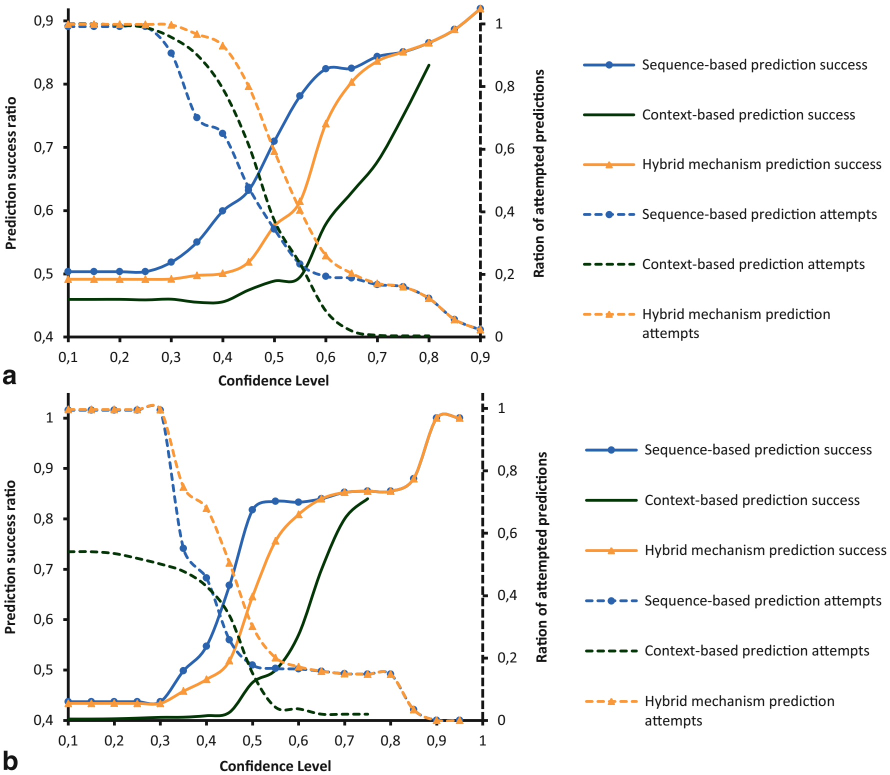

Fig. 23.6 Average prediction success and prediction attempt ratios for the three strategies for the **a** community-unaware and **b** community-assisted approaches

23.7 Conclusions and Future Work

The described framework is part of an open scalable service architecture and platform that aims to realise the concept of Pervasive Communities. A Pervasive Community demonstrates features related with context-awareness, self-organisation, self-improvement, and proactive behaviour in order to optimise and personalise the pervasive experience of an entire community (www.ict-societies.eu). Proactive behaviour is inherently related with user intent prediction, as for any system, to react appropriately for a given end-user, it needs to be aware of what that end-user is attempting to achieve. The presented approach aims to realise this by monitoring end-user behaviours and the context in which these behaviours occur. Observing temporal sequences of end-user actions and context cliques or snapshots can permit the discovery of past goals, and the prediction of future. This process is supported by knowledge originating from other community members. In a nutshell, the described framework aims to exploit recorded histories of users' interactions and context that are taking place in pervasive computing environments, in support of user intent prediction.

Future plans include further evaluation of the proposed mechanisms with additional datasets demonstrating diverse characteristics. To improve the performance of context-based user intent prediction, the adaptation and evaluation of alternative algorithms (such as Bayesian Networks) is planned. Regarding the community knowledge extraction, additional approaches will be evaluated such as the merging of individual user intent models. In addition, methods for the dynamic selection of action's escorting context will be researched. Currently, recorded context types are predefined and the same context escorts all actions. Selecting the attributes that better contribute on a per action base will improve prediction performance, but will also increase the volume of recorded data. In a similar manner, adapting dynamically the granularity of the discrete time-representation will improve the quality of prediction. Recorded timestamps can be refined in variable time intervals adapted to action and task duration.

References

Abowd, G.D., Bobick, I., Essa, I., Mynatt, E., Rogers, W.: The aware home: Developing technologies for successful aging. In: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, Canada, 28 July–1 Aug 2002

Adomavicius, G., Tuzhiin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE T. Knowl. Data En. **17**(6), 734–749 (2005)

Antwarg, L., Rokach, L., Shapira, B.: Attribute-driven hidden Markov model trees for intention prediction IEEE. Trans. Syst. Man. Cybern. C. **42**(6), 1103–1119 (2012)

Begleiter, R., El-Yaniv, R., Yona, G.: On prediction using variable order Markov models. J. Artif. Intell. Res. **22**(1), 385–421 (2004)

Doolin, K., Roussaki, I., Roddy, M., Kalatzis, N., Papadopoulou, E., Taylor, N.K., Liampotis, N., McKitterick, D., Jennings, E., Kosmides, P.: Societies: Where pervasive meets social. In: Alvarez, F., Cleary, F., Daras, P., Domingue, J., Galis, A., Garcia, A., Gavras, A., Karnourskos, S., Krco, S., Li, M.-S., Lotz, V., Müller, H., Salvadori, E., Sassen, A.-M., Schaffers, H., Stiller, B., Tselentis, G., Turkama, P., Zahariadis, T. (eds.) Future Internet Assembly Book. pp. 30–41. Springer, Heidelberg (2012)

Eagle, N., Pentland, A., Lazer, D.: Inferring social network structure using mobile phone data. In: International Workshop on Social Computing, Behavioral Modeling, and Prediction, Phoenix, Arizona, 1–2 April 2008

Gallacher, S., Papadopoulou, E., Taylor, N., Blackmun, F., Williams, H., Roussaki, I., Kalatzis, N., Liampotis, N., Zhang, D.: Personalisation in a system combining pervasiveness and social networking. In: Proceeding of 20th International Conference on Computer Communications and Networks, Hawaii, USA, 31 July-4 Aug 2011

Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project aura: Toward distraction-free pervasive computing. IEEE Pervasive Comput. **1**(2), 22–31 (2002)

Gopalratnam, K., Cook, D.J.: Online sequential prediction via incremental parsing: The active LeZi algorithm. IEEE Intell. Syst. **22**(1), 52–58 (2007)

Horvitz, E., Koch, P., Kadie, C.M., Jacobs, A.: Coordinate: Probabilistic forecasting of presence and availability. In: Proceeding of the 18th Conference on Uncertainty in Artificial Intelligence, Edmonton, Alberta, July 2002

Kalatzis, N., Liampotis, N., Roussaki, I., Kosmides, P., Papaioannou, I., Xynogalas, S., Zhang, D., Anagnostou, M.: Cross-community context management in cooperating smart spaces. Pers. Ubiquit. Comput. **18**(2), 427–443 (2014)

Magnusson, M.S.: Repeated patterns in behavior and other biological phenomena. In: Oller, K.D., Griebel, U. (eds.) Evolution of Communication Systems: A Comparative Approach, pp. 111–128. MIT Press, Cambridge (2004)

Ni, H., Abdulrazak, B., Zhang, D., Wu, S.: CDTOM: A context-driven task oriented middleware for pervasive homecare environment. Int. J. UbiComp. **2**(1), 34–53 (2011)

Roussaki, I., Kalatzis, N., Liampotis, N., Frank, K., Sykas, E.D., Anagnostou, M.: Developing context-aware personal smart spaces. In: Alencar, P., Cowan, D. (eds.) Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications, pp. 659–676. IGI Global, Hershey (2012)

Roussaki, I., Kalatzis, N., Liampotis, N., Kosmides, P., Anagnostou, M., Doolin, K., Jennings, E., Bouloudis, Y., Xynogalas, S.: Context-awareness in wireless and mobile computing revisited to embrace social networking. IEEE Commun. Mag. **50**(6), 74–81 (2012)

Singh, P., Williams, W.: LifeNet: a propositional model of ordinary human activity. In: Workshop on Distributed and Collaborative Knowledge Capture, Sanibel Island, FL, 23-26 Oct 2003

Sousa, J.P., Poladian, V., Garlan, D., Schmerl, B., Shaw, M.: Task-based adaptation for ubiquitous computing. IEEE. Trans. Syst. Man. Cybern. C. **36**(3), 328–340 (2006)

Tang, L., Liu, H.: Scalable learning of collective behavior based on sparse social dimensions. In: Proceedings of 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 Nov 2009

Thakor, M.V., Borsuk, W., Kalamas, M.: Hotlists and web browsing behaviour: An empirical investigation. J. Bus. Res. **57**(7), 776–786 (2004)

Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. 3rd edn. Morgan Kaufmann, Burlington (2011)