# Flow-Based Context Prediction

Stefan Föll, Klaus Herrmann, Christian Hiesinger
Institute of Parallel and Distributed Systems, Universität Stuttgart
Universitätsstrasse 38, 70569 Stuttgart, Germany
{stefan.foell,klaus.herrmann, christian.hiesinger}@ipvs.uni-stuttgart.de

## ABSTRACT

Context prediction has been recognized as an enabler for proactive pervasive services that anticipate future situations already ahead of time. Traditional context predictors are limited by their agnostic view on the targeted application domain when analysing context histories of past user behaviour. Awareness about the processes in which an entity is involved can provide rich information to foresee future context changes more accurately. We present an approach for context prediction in pervasive environments that are characterized by context-aware workflows. In order to benefit from the explicit knowledge about human behaviour in these environments, we devise a context predictor that learns the relationship of context changes with the flow of activities performed by humans. This relationship is encoded as a probabilistic state transition system that can be explored to determine the most likely paths of future context occurrences. Our evaluation shows that our enhanced predictor is able to extract patterns from context histories that are inaccessible to history-only predictors and significantly improves the prediction accuracy.

## Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]: Stochastic Processes, Probabilistic algorithm

## Keywords

Context prediction, Markov model, workflows, context awareness, probabilistic user behaviour

## 1. INTRODUCTION

Even though the initial vision of pervasive computing already dates back over a decade ago, the demand for sophisticated user-centric technology is still a major challenge in the focus of current research. Key for the development of pervasive technology which operates unnoticed from, but on behalf of the user, is seen in the principle of context awareness. While advances in recognition and processing of context have spawned a variety of new context-aware systems, these systems are often characterized by reactive behaviour and respond to changes in the current context *after* their actual occurrence in the real world. As a consequence, the level of intelligence found in these environments is restricted to what the user already can observe in his current situation. In order to render these environments more intelligent, applications should anticipate the future needs of users even *before* they physically appear [13]. As the human needs in pervasive scenarios are tightly coupled to their future context, sophisticated methods for *context prediction* are required to extend the temporal horizon of context awareness into the future. This enables the provision of proactive pervasive services that improve the experience of a user when interacting with his surroundings, applicable in relevant fields of pervasive computing such as for instance home automation, information recommendation, or human guidance in working places.

The most common approach for context prediction is to rely on context histories to deduce probable future context from past sequences of context data [11]. The idea is that from the analysis of context histories characteristics patterns in human behaviour can be discovered as a basis for prediction. For example, the next location to be visited by a user is typically extrapolated from the trajectory of his last locations. However, without any insight into the processes in which a human is embedded, context predictions are decoupled from the semantics of human behaviour and are susceptible to prediction errors. Many application domains of pervasive systems such as pervasive healthcare are characterized by human models of structured behaviour. These models exhibit the activities carried out by humans under varying conditions. In these scenarios, the context changes are predominantly motivated by the activities carried out by users, e.g., as user activities are associated with dedicated locations. Exploiting the knowledge about the processes in these environments provides the possibility to link context histories more accurately with user activities and enable predictions relevant to the current user situation.

Based on this motivation, we propose a new prediction scheme leveraging the concept of *Adaptable Pervasive Flows* (APFs) [9] that is in the focus of the European research project ALLOW[1]. APFs are context-aware workflows that model the activities of human entities. In contrast to tradi-



[1]This work is partially funded by the FP7 EU FET project Allow IST-324449

tional workflows, they are situated in the real world, as they are logically attached to moving humans, and context-aware, as they sense and synchronize with the changes in their context. In this paper, we show how to exploit flows as source of information to provide history-based predictors with domain knowledge for accurate context predictions. For this purpose, we introduce a novel flow-based predictor that relates context changes to states in human behaviour. Our predictor is based on a probabilistic state transition system that is learnt from the execution of flows and defines the search space for future context occurrences. For a prediction, we determine the most likely sequence of future context states reachable from the current user situation. As these search paths depend on the activities performed by humans, we are able to derive accurate context predictions that remain hidden for history-only approaches.

The paper is structured as follows. In Section 2 we describe the related work in context prediction. We then discuss context prediction based on Markov models in Section 3 and present a generic model of history-based predictors. In Section 4 we introduce Adaptable Pervasive Flows as a model of context-aware human behaviour. We then present our flow-based context predictor in Section 5. Section 6 shows the evaluation results in comparing history-based predictors with their flow-enhanced counterparts. Conclusion and future work are discussed in Section 7.

# 2. RELATED WORK

The most comprehensive approach to context prediction has been presented by Mayrhofer [12], who proposes a multi-layer system architecture for domain-independent context recognition and prediction. This work targets the prediction of high-level user context (context classes such as "in a meeting") which is derived from low-level context data (e.g. location, noise, etc.) in a preceding classification step. In order to improve the achieved prediction results, the author concludes that the inclusion of domain-specific knowledge would represent a promising approach, as addressed by our work. Sigg [16] directly predicts low-level context before deriving future high-level context in order to avoid information loss resulting from the aggregation of low-level context. However, this approach does not consider possible correlations among low-level context and deduces future context based on matched sequences from the past without taking advantage of domain-specific knowledge about human behaviour.

Furthermore, different algorithms have been proposed to allow for the prediction of specific categories of context. Most of the work in this area focuses on the prediction of user mobility from location histories [2], [17] [10], [1]. Beyond a history of location sequences, these approaches do not consider any further information for prediction. However, location predictors for wireless cellular networks can exploit the structure of geographic areas and direction information to anticipate future location changes more accurately [4]. In contrast to these approaches, we argue that the behaviour of humans is the most valuable information for prediction.

Moreover, the prediction of low-level user activities such as key pressings to improve the interaction with user interfaces has been studied [8], [6], [7]. Similar to location prediction, future user activities are derived from histories of past sequences. In our approach, activities are the constituent parts of context-aware workflows that synchronize with the behaviour of humans in the real world. The difference is that we use the knowledge provided by the workflows to predict additional context that evolves with the activities performed by users.

The idea to take advantage of domain knowledge in order to improve the recognition of hidden patterns from data is inherent to the field of syntactic pattern recognition [15]. Based on a statistical model that describes the generation of the data, patterns such as handwriting symbols or actions [3] can be discovered more accurately. Following this line of argumentation, we argue that also context prediction can benefit from a structural model of human behaviour that can be found in relevant fields of pervasive computing such as pervasive healthcare.

# 3. HISTORY-BASED PREDICTION

In the following section, we focus on history-based methods for context prediction. First, we will present the common procedure of history-based predictors. Then, we will analyse their shortcomings for prediction in flow-oriented environments.

## 3.1 Context Prediction

The rationale of context prediction is to extract characteristic patterns in human behaviour from histories of observed context data. As human behaviour cannot be captured exactly, the most common approach is to apply stochastic principles to describe the expected changes in user context. For this purpose, the occurrence of context (e.g. location) is regarded as a random variable $X$, which can be assigned values from a discrete set of context elements $C = \{id_1, id_2, .., id_n\}$. We assume that each context $c \in C$ can be associated with a unique symbolic identifier. For example, in terms of geographic positioning, symbolic location names such as "office" or "kitchen" provide a meaningful attribute of the user's location. Let the context history $H = c_1, c_2, ..., c_n$ be defined as a sequence of context elements $c_i \in C$ ordered according to their time of occurrence. Sequential changes in context can thus be considered as a stochastic process $\chi$ that describes the evolution in user behaviour with distribution $P(X_1 = c_1, X_2 = c_2, ..., X_n = c_n)$.

The most widely employed history-based predictors from related work [2, 7, 17] are based on discrete Markov processes. The Markov assumption is inherent to two different classes of predictors - the fixed order *0(k)* Markov predictors and the predictors based on varying order Markov models. $0(k)$ Markov predictors consider a fixed window of past context observations for prediction. The order $k$ of the Markov predictor determines the length of the window that influences the predicted context. Consequently, the part of the history relevant for prediction is given by the $k$ last observations $H(k) = c_{n-k+1}, ..., c_n$. Assuming a stationary stochastic process, the conditional probability distribution can be estimated from the occurrence of context changes in the history. The prediction is then determined by the context $c_{n+1}$ which has most frequently followed the sub-sequence $H(k)$ in the entire history $H$. The restriction of a fixed order is relaxed by so-called Markov Models of varying orders. The most popular varying order Markov predictors are based on the data compression algorithm of Ziv and Lempel [19].

In order to incorporate domain knowledge, our prediction scheme is based on a generic model of history-based predictors. We extend this generic model in Section 5 to combine it with knowledge about user activities. A very important consequence of this technique is that it allows us to apply our approach to several variants of state-of-the-art context predictors. In our generic model, we define a history-based predictor as a probabilistic state transitions system. The definition captures the common nature of Markov predictors: A state is a sequence of one or more past context elements upon which predictions are derived from the analysis of context histories.

**Definition 1** (History-based context predictor): A history-based context predictor $\hat{P}$ is specified by a probabilistic state transition system $(S, C, \delta, p)$, where

- $C$ is the set of discrete context elements
- $S$ denotes the set of history states
- $\delta : S \times C \rightarrow S$ denotes the transition function that describes possible context changes
- $p : S \times C \rightarrow [0, 1]$ indicates the probability for a specific context change

Each $s \in S$ corresponds to a history state. When observing a context $c \in C$ during state $s$, the history state changes to $s' = \delta(s, c)$. Thus, the history state evolves with new context observations. However, the transition function is partial as not necessarily each context can be observed during a history state. The predictor encodes the probability of future context occurrences in transition probabilities. The probability $P(c|s)$ to expect a context $c \in C$ depends on the history state $s$ and is indicated by $p(s, c)$. The sum of all probabilities over all outgoing transition has to be one, i.e., $\forall s \in S : \sum_{\forall c \in C : \delta(s,c) \neq \emptyset} p(s, c) = 1$. Initial states, as known from classical automaton theory, are defined by the current history state at time of prediction.

## 3.2 Analysis of History Restrictions

The limiting factor of history-based predictors is implied by their nature – their dependency on past observations as the only indicator to what can follow next. Due to the Markov assumption, the sequence of past context occurrences must carry enough information to make accurate predictions. However, historical information may not be sufficient for enabling accurate predictions in every case. This observation is especially relevant in cases where the situation of a user is defined in terms of higher-level behaviour that more precisely implies the next context to occur. The accuracy of predictions is low if the next context has a semantic association with the user's behaviour that cannot be learnt from the history. If we imagine for example a worker leaving his office, then there are many equally likely options for his future location according to the information from the location history. If we could access the knowledge that the worker decided to visit a customer, we could use the domain insight to forecast his next location more accurately. Here, the capabilities of a history predictor are very limited, because the knowledge from the history is not able to capture this form of hidden information.

Taking the user behaviour into account, we are able to provide history predictors with the necessary knowledge to reduce ambiguities. As a consequence, we require a model of human behaviour that allows us to interpret situations,

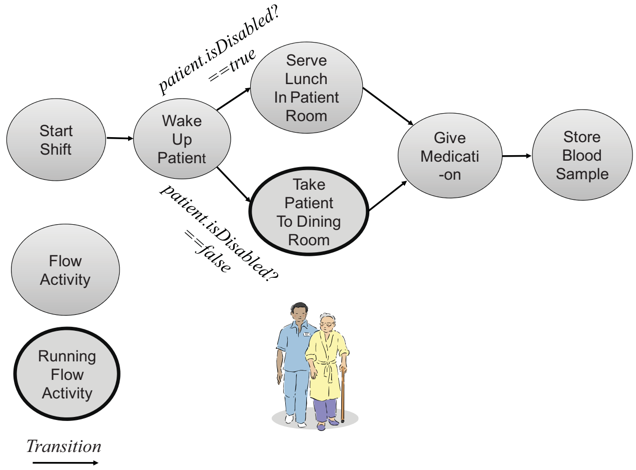


**Figure 1: Representation of a Pervasive Flow attached to a Nurse**

based on a human's past and future activities and the context under which the activities are taking place. For this purpose, we will leverage on *Adaptable Pervasive Flows* as a context-aware model of human behaviour as described in the next section.

# 4. ADAPTABLE PERVASIVE FLOWS

Workflows are often inherent to human behaviour in the real world – either explicitly or implicitly. Explicitly, workflows can be found in domains such as hospitals or logistics, where humans follow common procedures in daily routines. In these domains, regular procedures are carried out as best practices or obligations by the personnel. Frequently, the worfklows are defined by underlying business processes and are readily available as business process models. Moreover, even in less obvious daily situations, workflows are present and or can be discovered with suitable mining techniques [5, 18]. For example at home during cooking, at sports or when interacting with electronic appliances humans people behave in a structured way.

Based on this motivation, Adaptable Pervasive Flows (also simply called *flows* hereafter) [9] have been proposed as a model for human-oriented pervasive applications. Flows are context-aware workflows that are situated in the real world. They describe the activities of a human and adapt to changes in the context of the human's environment. Whereas [9] presents the opportunities of flows for adapting pervasive applications, we exploit flows as providers of domain-specific knowledge for context prediction in this work. The knowledge stems from the fact that a) a flow provides insight in the current state of its associated entity b) a flow models paths of future activities. For this purpose, we extend the work of [9] with a generic flow-based context prediction scheme.

## 4.1 Flow Model

For the purpose of context prediction, we do not rely on a specific technology and focus on the generic model of flows. A *flow model* describes the activities of a human under changing contextual conditions. As an example consider the flow attached to a nurse in a hospital shown in Figure 1. During her work day, a nurse carries out regular activities such as "give medication" or "serve lunch". After the start of her shift she progresses in her workflow and executes activities that are associated with contextual conditions. For example, the concrete activities of a nurse depend on the health conditions of the patient she is caring for.

**Definition 2** (Flow Model): A Flow Model $f$ is specified by a directed graph $(A, E, P(C), t)$, where

- $A$ denotes the set of *activities*
- $E \subseteq A \times A$ defines a *control flow*
- $P(C)$ is the set of predicates over the *entity context* $C$
- $t : E \to P(C)$ associates each control link with a transition condition

The flow model defines a *control flow* over the set of flow *activities* based on a directed graph. Each flow contains a start activity from which there is a path to any other activity in the flow. The control flow constraints the possible paths of activity executions. An activity path $a_1 \to a_2, ..., a_{n-1} \to a_n$ in the flow consists of pairs of connected activities, i.e., $(a_i, a_{i+1}) \in E$. An activity can only be executed, if one of its predecessor has been completed. The completion of activities is triggered by conditions that are checked at run-time. These conditions are related to the context of a human (also more generally called *entity*) in its current situation. For this purpose, context recognition techniques such activity sensing or other sources of information (e.g. patient data) are used. Through the integration of context information, the flow is synchronized with the real-world behaviour of humans.

## 4.2 Flow Instance

The run-time representation of a flow is referred to as *flow instance*. Flow instances are created in a context-aware manner based on contextual triggers. For example, as soon as a nurse starts her shift, a flow instance is created and attached to her. A flow instance exposes the state of a flow, which dynamically evolves during its lifetime.

**Definition 3** (Flow State): The state of a flow $f$ is specified by its currently running activity, which is given by the function $state : f \mapsto a \in A$.

The state of a flow instance is controlled by a *flow engine*. A flow engine runs flows and accesses the context information that influences their states. The state of a flow is always well defined, i.e., there is no state in between two activities. If a successor activity starts, the preceding activity is terminated. Thus, a flow state is active for a certain period of time.

# 5. FLOW-BASED CONTEXT PREDICTION

In this section, we introduce a new *Flow Predictor* that combines both sources of knowledge – context histories *and* flows – to leverage the additional information present in flows. The relationship of flows and context is encoded as a probabilistic state transition system that includes activity information. To express this relationship, we need to extend the history-based model for context prediction.

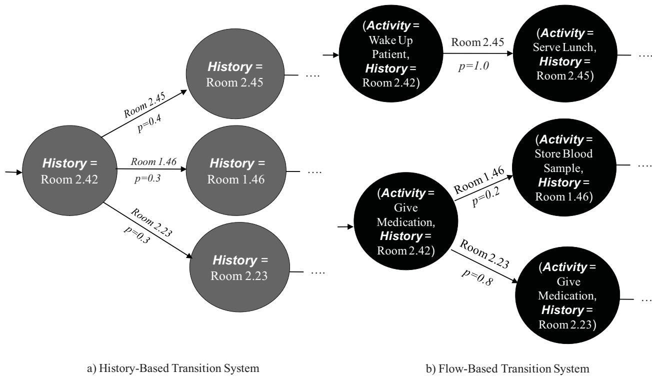


**Figure 2: History- vs. Flow-based Transition Systems**

**Definition 4** (Flow Predictor): A flow predictor $\hat{P}_f$ is associated with a flow $f$ and is based on a history predictor $\hat{P}$. $\hat{P}_f$ is formally defined as probabilistic state transition system $(\hat{S}, C, \tau, p, f)$, where

- the states $\hat{S} = (S \times A)$ are the Cartesian product of the states of the history-based predictor $\hat{P}$ and activities of flow $f$
- $C$ is the set of discrete context elements
- $\tau \subseteq \hat{S} \times C \times \hat{S}$ denotes the transition relation
- $p : \tau \to [0, 1]$ indicates the transition probability with $\forall s \in \hat{S} : \sum_{\forall c \in C, s' \in \hat{S}:(s,c,s') \in \tau} p(s, c, s') = 1$

States are now defined as tuples of flow activities and history states. Thus, we establish a relation among both. The flow activities in the flow predictor introduce a new differentiating criterion for history states. We can now find the *same* history state in different states of the flow predictor. Each of these history states may be associated with distinct transition probabilities. This enables accurate predictions tailored to the current user activity. In contrast, a history-based predictor represents each history state only once.

Figure 2 illustrates this difference for the prediction of the nurse's location. The history-based predictor (left side) contains three possibilities for changing from "$Room2.42$" to other locations. All of these locations are almost equally probable. In contrast to this, the flow predictor (right side) links locations to activities. Note that the state relating to "$Room2.42$" has been split in two states, each being associated with a different activity ("$GiveMedication$") and ("$WakeUpPatient$"). Using this activity information, the flow predictor can make a much more reliable prediction: When the nurse is executing the activity "$GiveMedication$" in "$Room2.42$", only two out of the three locations are likely to be visited, and the highest probability associated with an outgoing transition has increased over the history-based predictor. Moreover, if the nurse is executing "$WakeUpPatient$" in the same location "$Room2.42$", then only one possibility remains for the next location ("$Room2.45$"). Thus, a combination with activity information extracted from flows splits a single history state into multiple states such that for each of the following states the probability may increase.

## 5.1 Online Learning from Flow-Enhanced Context Histories

The flow predictor provides the formal framework for incorporating user activity information into context predictions. However, the transition system underlying the flow predictor has to be learnt from the execution history of flows. The goal of learning is to obtain the state transitions and associated probabilities that accurately describe the evolution of the entity's context in the target domain.

**Algorithm 1** Online Learning

**Require:** Flow $f$
1: $\tau \leftarrow \{\}, buffer \leftarrow \{\}$
2: $h \leftarrow \epsilon, a_n \leftarrow state(f)$
3: **while** true **do**
4: $\quad e \leftarrow nextElementFromHistory(H_f)$
5: $\quad$ **if** $e \in A$ **then**
6: $\qquad buffer \leftarrow buffer \cup \{a_n\}$
7: $\qquad a_n \leftarrow e$
8: $\qquad$ **if** $h \neq \epsilon$ **then**
9: $\qquad\quad \hat{S} = \hat{S} \cup \{(h, a_n)\}$
10: $\qquad$ **end if**
11: $\quad$ **end if**
12: $\quad$ **if** $e \in C$ **then**
13: $\qquad h' \leftarrow \delta(h, e)$
14: $\qquad$ **if** $h \neq \epsilon$ **then**
15: $\qquad\quad \hat{S} = \hat{S} \cup \{(h', a_n)\}$
16: $\qquad\quad \tau \leftarrow \tau \cup \{((h, a_n), e, (h', a_n))\}$
17: $\qquad\quad$ increment $count((h, a_n), e, (h', a_n))$
18: $\qquad\quad$ **for all** $a_{past} \in buffer$ **do**
19: $\qquad\qquad \tau \leftarrow \tau \cup \{((h, a_{past}), e, (h', a_n))\}$
20: $\qquad\qquad$ increment $count((h, a_{past}), e, (h', a_n))$
21: $\qquad\quad$ **end for**
22: $\qquad$ **end if**
23: $\qquad h \leftarrow h'$
24: $\qquad buffer \leftarrow \{\}$
25: $\quad$ **end if**
26: **end while**

The information used for learning is a sequence of changes in either the flow state or the context. This information is stored in the so-called *flow-enhanced history*, denoted by $H_f$. The changes in flow activities and context appearing in the real world (and thus also in $H_f$) can be arbitrarily interleaved. Thus, the flow-enhanced history is defined as $H_f = c_0, a_1, c_1, a_2, ..., c_n$ with $c_i \in C \cup \{\epsilon\}, a_j \in A \cup \{\epsilon\}$ where $\epsilon$ denotes the empty symbol. $H_f$ is a sequential stream of events to which each new observation is appended at run-time. This is necessary, since our predictor is run in an on-line manner, i.e., the training phase is executed simultaneously to the execution of the flow-based system at runtime. The probability for future context occurrences are estimated from the observed frequencies in $H_f$. For this purpose, we associate a counter with each transition $t = ((h, a), c, (h', a')) \in \tau$, denoted as *count*$(t)$, which stores the frequency of past transitions.

Algorithm 1 shows the procedure executed for online learning: Each new element of $H_f$ can be either a new activity or a new context. We insert new transitions in the predictor only if a context change happens. If the last added state is $(h, a_n)$ and a new context $c$ is observed, we insert a transition $((h, a_n), c, (h', a_n))$ (line 16). The representation of $h$ and $h'$ depends on the underlying Markov model. For example, in case of a $\mathcal{O}(1)$ Markov model, the new history state is defined only by the observed context $c$, i.e., $h' = c$. In contrast, for higher order or varying order Markov models further past context observations contribute to the new history state $h'$. Since an arbitrary series of activity changes may happen in $H_f$ before the next context change to $c$ occurs, we buffer these activities (line 6). Once, the new context $c$ occurs, we also add a transition $((h, a_{past}), c, (h', a_n))$ for each of these buffered past activities (lines 18-21) since for each state $(h, a_{past})$ the next context is $c$. Suppose, for example, that the current predictor state is $s_1 = (history = "Room2.42", activity = "GiveMedication")$ and the next elements in $H_f$ are $"StoreBloodSample"$ (activity change) and $"Room2.43"$ (context change). Then $s_1$ *and* the state $s_2 = (history = "Room2.42", activity = "StoreBloodSample")$ should be related to the next context and we insert a transition to state $s_3 = (history = "Room2.43", activity = "StoreBloodSample")$ from both $s_1$ and $s_2$. After this step, the buffer is emptied and we wait for next observed event.

The probability of a transition $t = ((h, a), c, (h', a')) \in \tau$ can be derived as a relative frequency measure from the transition counters:

$$p((h, a), c, (h', a')) = \frac{count((h,a),c,(h',a'))}{\sum_{c' \in C} \sum_{a'' \in A} count((h,a),c',(\delta(h,c'),a''))}$$

For the calculation of $p((h, a), c, (h', a'))$, we take all outgoing transitions from the state $(h, a)$ into account. The probability is derived from the transition counters whenever a prediction has to be made. Due to the representation of activity information, there is an increased cost in storage associated with the flow predictor. The state space is now of size $\mathcal{O}(|S| \cdot |A|)$, since activities are combined with the context from histories. Consequently, also the encoding of transitions requires more space and has complexity $\mathcal{O}(|S| \cdot |C| \cdot |A|^2)$. In the scenarios addressed by this work, the cost will be affordable, as we assume a limited set of activities to be of interest and the real cost to be significantly below the worst case estimation, as activities are not observable at each context.

## 5.2 Calculation of Predictions

For predicting future context we distinguish between two classes of prediction - *short-term* and *long-term* prediction. In both cases we are interested in future context occurrences, that follow the current context history $H$. Let $c_n$ denote the last context observed from $H$. For short-term prediction the goal is to determine the context $c_{n+1}$ that will most probably occur next. Long-term prediction extends the time horizon to more distant points in the future. For this purpose, we define the number of future occurrences as *prediction horizon*. Formally, for a prediction horizon of $h$, the goal is to identify the most probable sequence of future context elements $c_{n+1}, c_{n+2}, ..., c_{n+h}$. In the following, we describe the algorithmic approach to calculate these predictions based on the flow predictor.

### 5.2.1 Short-term prediction

The starting point for short-term prediction is given by the current predictor state $(h, a)$, from which the transition system is traversed. Algorithm 2 shows the steps involved in the calculation of the most probable next context.

**Algorithm 2** Short-Term Context Prediction

```
Require: current history state h
Require: current flow state a = state(f)
Ensure: c_{n+1} = arg max_{c∈C} P(X_{n+1} = c|(h, a))
1: for all c ∈ C do
2:     Prob(c) ← Σ_{(h',a')∈Ŝ} p((h, a), c, (h', a')}
3: end for
4: return arg max_{c∈C} Prob(c)
```

**Require:** current history state $h$
**Require:** current flow state $a = state(f)$
**Ensure:** $c_{n+1} = \arg\max_{c \in C} P(X_{n+1} = c|(h, a))$
1: **for all** $c \in C$ **do**
2: $\quad Prob(c) \leftarrow \sum_{(h',a') \in \hat{S}} p((h, a), c, (h', a')\}$
3: **end for**
4: **return** $\arg\max_{c \in C} Prob(c)$

For short-time prediction, we have to take into account that the same context $c$ may be reached via different transitions from $(h, a)$. Therefore, we have to sum the probabilities associated with each transition that is labelled with context $c$ (line 2). Finally, the context returned as the prediction is the one with maximum probability, i.e., $c_{n+1} = \arg\max_{c \in C} P(X_{n+1} = c|(h, a))$ (line 4). The worst case time complexity is $\mathcal{O}(|A| \cdot |C|)$ since the next context may potentially occur in each of the flow activities. However, we stress that, in practice, the search space is much more restricted by the flow structure. This structure only allows for a small subset of all activity-context combinations.

### 5.2.2 Long-term prediction

The calculation of most likely paths is known from Hidden Markov Models (HMMs) [14] where the Viterbi algorithm is used to discover paths of so-called hidden states for given observations. However, HMMs are based on predefined sets of states and constant transition probabilities. In our case, transition probabilities vary for each prediction horizon, and future paths depend on the initial state at the time of prediction and need to be explored. We address these issues in Algorithm 3.

The algorithm is based on an iterative approach that calculates the most likely path (sequence of context occurrences) of length $h$ (prediction horizon). It starts with path length 1 and determines the most likely path of length $i$ ($1 < i \leq h$) based on paths of length $i - 1$ (line 2-9). For this purpose, we compute $Prob_i(c)$, which denotes the probability of the most likely path of length $i$ that ends in the occurrence of context element $c$. Initially, we set $Prob_0(c) = 1$ and, for all $i > 0$, $Prob_i(c)$ is calculated by adding the transition probabilities that reach $c$ from paths of length $i-1$ (line 4). Based on $Prob_i(c)$, the sequence of context elements that define the most likely path is built incrementally. For this purpose, we associate with $path_i(c)$ the most likely sequence of context occurrences that ends in $c$ for a path of length $i$. We then append to $path_i(c)$ the context that maximizes the path probability in each iteration (line 6), starting from a path length of $i = 1$.

For these calculations, we first define $reachable_k(h, a)$ as being the set of all predictor states reachable from state $(h, a)$ through an arbitrary path of length $k$. That is, we initially have $reachable_0(h, a) = \{h, a\}$ and for all $k > 1$, $reachable_k(h, a)$ can be explored by following the outgoing transitions. Based on this, we define $R_k(c)$ as being the set of predictor states reachable through context $c$ from any state in $reachable_{k-1}(h, a)$. More formally, we write

$$R_k(c) = \{(h', a') \in \hat{S} | \exists (h'', a'') \in reachable_{k-1}(h, a) : ((h'', a''), c, (h', a')) \in \tau\}$$

**Algorithm 3** Long-Term Context Prediction

**Require:** current history state $h$
**Require:** current flow state $a = state(f)$
**Require:** prediction horizon h
**Ensure:** $c_{n+1}, c_{n+2}, ..., c_{n+h}$ most likely path
1: $i \leftarrow 1$
2: **while** $i <= h$ **do**
3: $\quad$ **for all** $c \in C$ **do**
4: $\quad\quad Prob_i(c) = max_{c' \in C}\{Prob_{i-1}(c') \cdot \sum_{(h,a) \in R_{i-1}(c'),(h',a') \in \hat{S}} p((h, a), c, (h', a')\}$
5: $\quad\quad$ **if** $(i > 1)$ **then**
6: $\quad\quad\quad path_i(c) \leftarrow$ append $\arg\max_{c' \in C}\{Prob_{i-1}(c') \cdot \sum_{(h,a) \in R_{i-1}(c'),(h',a') \in \hat{S}} p((h, a), c, (h', a')\}$
7: $\quad\quad$ **end if**
8: $\quad$ **end for**
9: $\quad i \leftarrow i + 1$
10: **end while**
11: $c^* = \arg\max_{c \in C} Prob_h(c)$
12: $path_h(c^*) \leftarrow$ append $c^*$
13: **return** $path_h(c^*)$

with $(h, a)$ being the current state at which the prediction starts. $R_k(c)$ is used in the algorithm (line 4 and 6) to identify the context occurrences that lie on the most likely paths to $c$ for each step of the iteration.

After the termination of the iteration, the most likely path for horizon $h$ is the path that maximizes the path probability for a context $c^* \in C$. Consequently, we can return the stored path associated with the context $c^* = \max_{c \in C} Prob_h(c)$ (line 9). As the path ends with $c^*$, we have to append this context to the complete path up to horizon $h$. The iterative approach guarantees that the time complexity is bound by $\mathcal{O}(h * (|A| \cdot |C|)^2)$ in the worst case. However, the search space will in reality be often restricted by the fact that not all context and activities are reachable from the current state.

## 6. EVALUATION

We have implemented a simulation environment in order to evaluate the suitability of Adaptable Pervasive Flows for context prediction. We compare history-only predictors with their flow-enhanced counterparts based on synthetic context histories. This allows us to analyse the accuracies of the predictors for a spectrum of possible scenarios. We have implemented the *0(k)* family of Markov predictors as well as the flow enhanced-version of these. Although our approach is applicable to any form of discrete context, we study the accuracy of location prediction based on an activity-based mobility model that associates activities with locations as explained in the following.

First, we randomly generate flow models of different structure and size. Flow models are created from two different workflow patterns, i.e., sequences and branches, and form directed acyclic graphs. Second, we probabilistically associate each flow activity with locations from the domain $L = \{l_1, l_2, ..., l_n\}$. For each activity, we independently derive location visit probabilities based on a Zipf distribution, so that the probability to visit the $i$-th location during an activity is given by $P(X = i) = \frac{i^{-s}}{\sum_{n=1}^{|L|} n^{-s}}$. The exponent $s$ allows us to vary the density of the visit probabilities. Based on the activity-based model of user mobility, we gen-

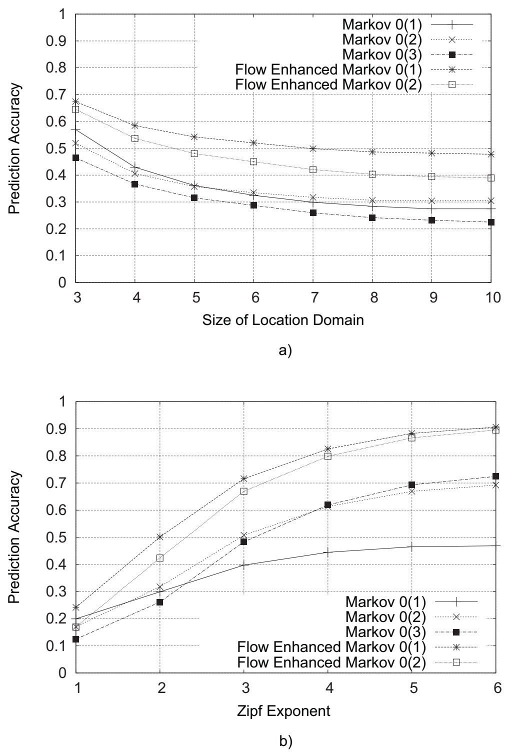


**Figure 3: Short-Term Prediction Accuracies for Activity-Based Mobility Model with Parameter Settings a) Zipf exponent $s$=2 b) Location Domain $|L| = 7$**

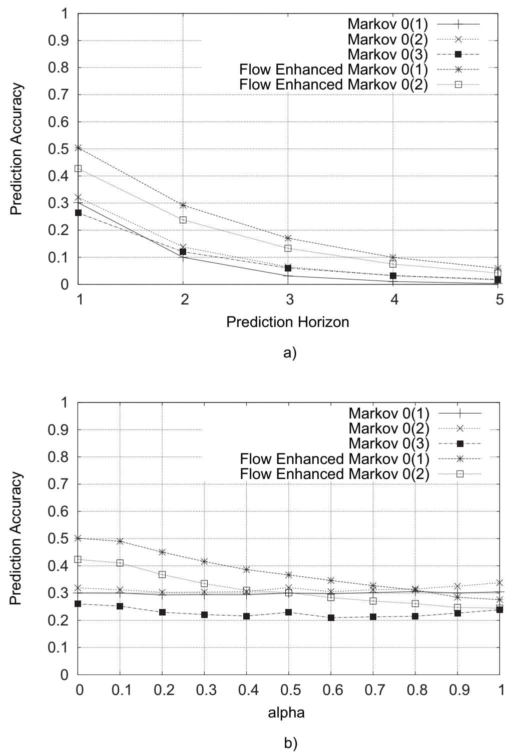


**Figure 4: a) Long-Term Prediction Accuracies b) Prediction Accuracy for History-Generated Mobility Patterns**

erate $n$ flow-enhanced context histories $H_{f_1}, H_{f_2}, ..., H_{f_n}$ as sequential input for the predictors. We compare the different predictors based on an accuracy metrics, that is defined as the ratio of number of correct predictions to all predictions made. If a prediction is not possible due to the fact that the current context has not been learnt before, we count it as incorrect prediction. Predictions are determined simultaneously to the learning phase, i.e., after each predictor update we compute a prediction and validate it. A simulation run consists of a generated flow for which we create 100 context histories that describe possible executions of the flow. The results discussed in the following represent the average of 500 simulation runs for each measurement.

Figure 3 a) shows the short-term prediction accuracies for an increasing size of the location domain and Zipf exponent $s = 2$. Due to the higher uncertainty associated with a larger location domain, the prediction accuracy is negatively affected for all predictors. However, the flow-enhanced predictors outperform the history-based predictors for all of the evaluated sizes of $L$. Particularly, the relative improvement rises from 19% to 56% compared to the best history-based predictor for an increasing size of $L$. This illustrates the capability to resolve ambiguities from the history due to the available flow knowledge. Since we associate a single location visit with an activity in the simulation, higher-order Markov models do not improve the accuracy of the flow-enhanced predictors. As more states have to be learnt in this case, additional prediction misses are caused.

In Figure 3 b) we compare the predictors for a location domain of $|L| = 7$ and varying Zipf exponents. For increasing exponents the locations visit probabilities exhibit a highly skewed distribution, so that the predictors are able to deduce a higher fraction of correct predictions. However, the flow-enhanced predictors are able to capture patterns that remain hidden for the history-based predictors. The flow-enhanced predictors achieve a relative improvement of 25% in prediction accuracy compared to the best history-based predictor. Moreover, the results show that history-based Markov models can benefit from a larger memory for prediction if activities are more restricted to specific locations. In this case, longer sequences of past locations more accurately imply the next location. Nevertheless, still a substantial fraction of patterns can only be distinguished with flow knowledge.

Figure 4 a) depicts the results of long-term prediction for parameters of $|L| = 7$ and Zipf exponent 2. The absolute prediction accuracies for all predictors naturally decrease for higher prediction horizons. Particularly, for horizon 5 the absolute accuracy has reached a degree, where no sensible predictions can be made any more. However, the relative improvement in accuracy of flow-based prediction compared to the best history predictor monotonically increases from 57 % for horizon 1 to considerable 331 % for horizon 5. Consequently, especially long-term predictions can benefit from our enhanced context prediction. In the next step, we extend the simulation model with the possibility to include history-generated patterns in the context histories. The history patterns are generated based on a $0(2)$ Markov source that is trained from the location traces of the activity-based model. This model naturally favours history-based predictors due to the underlying Markov assumption. We introduce the parameter $\alpha$ that allows to vary between both models, i.e., $\alpha$ indicates the portion of the context history which is generated by the Markov source and the portion $(1-\alpha)$ which adheres to the activity-based model. Figure 4 b) shows a monotonic decrease of the prediction accuracies of the flow-based predictors for increasing values for $\alpha$, while the history-based predictors remain constant. For $\alpha \geq 0.8$ the best flow-based predictor even performs worse than the *0(1)* Markov predictor. The reason is that, due to the correlation with flow activities, history-based patterns are scattered over many states of the flow predictor. The consequence is that the patterns cannot be learnt as fast as in the case of a classical history predictor, and more training data is necessary to achieve the same accuracy. We will address this issue in future work by the design of a hybrid prediction scheme, that involves components of both predictors and only utilizes flow knowledge for patterns that cannot be discovered by history-based predictors.

# 7. CONCLUSION

We have presented a new context prediction scheme that is able to provide history predictors with domain-specific knowledge inherent to flow-oriented pervasive environments. The domain-specific knowledge arises from a model of pervasive applications that describes the activities of pervasive users as context-aware workflows. Our enhanced context predictor learns the relationship of flow activities with context changes observed in the real world. We represent this relationship as a probabilistic state transition system which is incrementally refined from the execution of flows at runtime. For context prediction we traverse the state space of possible context changes to determine the most likely paths of future context occurrences. In our evaluation, we have shown that the inclusion of knowledge about user activities in the prediction model significantly improves the prediction accuracy, as classical predictors are limited by their agnostic view on the application domain.

In future work, we will validate the benefits of flow-based context prediction in real world evaluation studies. For this purpose, we will record the workflows of medical personnel in a hospital and evaluate the accuracy of our prediction algorithm for real context data. Moreover, we will extend our predictor to deal with time-varying patterns in human behaviour. Therefore, we will relax our current assumption of context changes characterized by strong stationary properties and discard past knowledge that is negatively influencing the prediction.

# 8. REFERENCES


[1] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, M. Kyriakakos, and A. Kalousis. Predicting the location of mobile users: a machine learning approach. In *Proc. of the 6th Intl. Conf. on Pervasive Services*, 2009.

[2] A. Bhattacharya and S. K. Das. Lezi-update: An information-theoretic approach to track mobile users in pcs networks. In *Proc. of the 5th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking*, 1999.

[3] A. F. Bobick and Y. A. Ivanov. Action recognition using probabilistic parsing. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.

[4] C. Cheng, R. Jain, and E. van den Berg. *Handbook of Wireless Internet*, chapter Location prediction algorithms for mobile wireless systems. CRC Press, 2003.

[5] J. E. Cook and A. L. Wolf. Automating process discovery through event-data analysis. In *Proc. of the 17th Intl. Conf. on Software Engineering*, 1995.

[6] B. D. Davison and H. Hirsh. Predicting sequences of user actions. In *Workshop on Predicting the Future: AI Approaches to Time Series Analysis*, 1998.

[7] K. Gopalratnam and D. J. Cook. Online sequential prediction via incremental parsing: The active lezi algorithm. *IEEE Intelligent Systems*, 22:52–58, 2007.

[8] M. Hartmann and D. Schreiber. Prediction algorithms for user actions. In *In Proc. of Intl. Conf. on Adaptive Business Information Systems*, 2007.

[9] K. Herrmann, K. Rothermel, G. Kortuem, and N. Dulay. Adaptable pervasive flows - an emerging technology for pervasive adaptation. In *Proc. of the Workshop on Pervasive Adaptation at the 2nd Intl. Conf. on Self-Adaptive and Self-Organizing Systems*, 2008.

[10] D. Katsaros and Y. Manolopoulos. Prediction in wireless networks by markov chains. *IEEE Wireless Communications*, 16:56–63, 2009.

[11] R. Mayrhofer. Context prediction based on context histories: Expected benefits, issues and current state-of-the-art. In *Proc. of the 1st Intl. Workshop on Exploiting Context Histories in Smart Environments*, 2005.

[12] R. M. Mayrhofer. *An Architecture for Context Prediction*. PhD thesis, Johannes Kepler University of Linz, 2004.

[13] M. M. P. Nurmi and J. A. Flanagan. Enabling proactiveness through context prediction. In *In Proc. of the 2nd Workshop on Context Awareness for Proactive Systems*, 2005.

[14] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[15] R. Schalkoff. *Pattern Recognition. Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc, 1992.

[16] S. Sigg. *Development of a novel context prediction algorithm and analysis of context prediction schemes*. PhD thesis, University of Kassel, 2008.

[17] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Transactions on Mobile Computing*, 5:1633–1649, 2004.

[18] W. M. P. van der Aalst, B. F. van Dongen, L. Maruster, G. Schimm, and A. J. M. M. Weijters. Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering*, Volume 47:237–267, 2003.

[19] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24:530–536, 1978.