A continuous-time model-based approach to activity recognition for ambient assisted living

Laura Carnevali[1], Christopher Nugent[2], Fulvio Patara[1], and Enrico Vicario[1]

[1] Department of Information Engineering, University of Florence, Italy,
{laura.carnevali,fulvio.patara,enrico.vicario}@unifi.it
[2] School of Computing and Mathematics, University of Ulster, United Kingdom
cd.nugent@ulster.ac.uk

Abstract. In Ambient Assisted Living (AAL), Activity Recognition (AR) plays a crucial role in filling the semantic gap between sensor data and interpretation needed at the application level. We propose a quantitative model-based approach to on-line prediction of activities that takes into account not only the sequencing of events but also the continuous duration of their inter-occurrence times: given a stream of time-stamped and typed events, online transient analysis of a continuous-time stochastic model is used to derive a measure of likelihood for the currently performed activity and to predict its evolution until the next event; while the structure of the model is predefined, its actual topology and stochastic parameters are automatically derived from the statistics of observed events. The approach is validated with reference to a public data set widely used in applications of AAL, providing results that show comparable performance with state-of-the-art offline approaches, namely Hidden Markov Models (HMM) and Conditional Random Fields (CRF).

Keywords: Ambient Assisted Living (AAL), Activity Recognition (AR), continuous-time stochastic models, transient analysis, on-line prediction, process enhancement.

1 Introduction

Ambient Assisted Living (AAL) aims at providing assistance to people living within smart environments through the integration and exploitation of new sensing technologies, data processing techniques, and services [11]. To this end, Activity Recognition (AR) plays a crucial role in filling the gap between sensor data and high level semantics needed at the application level [18]. This comprises a major ground for the application of quantitative approaches to diagnosis, prediction, and optimization.

A large part of techniques applied for AR [30] rely on or compare with Hidden Markov Models (HMM) [27]: the current (hidden) activity is the state of a Discrete Time Markov Chain (DTMC), and the observed event depends only on the current activity; stochastic parameters of the model can be determined through supervised learning based on some given statistics; and efficient algorithms are finally available to determine which *path* along hidden activities may have produced with maximum likelihood a given *trace* of observed events. While the ground truth is often based on data sets annotated w.r.t. predefined activities [27, 19], more data driven and unsupervised approaches have been advocated where activities are identified through the clustering of emergent recurring patterns [20, 3]. Various extensions were proposed to encode memory in HMM by representing sojourn times through discrete general or phase type distributions [16]. However, also in these cases, the discrete-time abstraction of the model prevents exploitation of continuous time observed between event occurrences.

To overcome this limitation, [8] proposes that the evolution of the hidden state be modeled as a non-Markovian stochastic Petri Net emitting randomized observable events at the firing of transitions. Approximate transient probabilities, derived through discretization of the state space, are then used as a measure of likelihood to infer the current hidden state from observed events. To avoid the complexity of age memory accumulated across subsequent states, the approach assumes that some observable event is emitted at every change of the hidden state. Moreover, the structure of the model and the distribution of transition durations are assumed to be given.

Automated construction of an unknown model that can accept sequences of observed events is formulated in [24] under the term of *process elicitation*, and solved by various algorithms [26] supporting the identification of an (untimed) Petri Net model. Good results are reported in the reconstruction of administrative workflows [25], while applicability appears to be more difficult for less structured workflows, such as healthcare pathways [15]. As a part of the process mining agenda, *process enhancement* techniques have been proposed to enrich an untimed model with stochastic parameters derived from the statistics of observed data [22].

In this paper, we propose a quantitative model-based approach to on-line prediction of activities that takes into account not only the type of events but also the continuous duration of activities and of inter-events time. Given a stream of time-stamped and typed events, transient probabilities of a continuous-time stochastic model are used to derive a measure of likelihood for on-line diagnosis and prediction of performed activities. Transient analysis based on transient stochastic state classes [12] maintains the continuous-time abstraction and keeps the complexity insensitive to the actual time between subsequent events. While the structure of the model is predefined, its actual topology and stochastic parameters are automatically derived from the statistics of observed events. Applicability to the context of AAL is validated by experimenting on a reference annotated data set [27], and results show comparable performance w.r.t. offline classification based on Hidden Markov Models (HMM) and Conditional Random Fields (CRF).

In the rest of the paper: the data set of [27] is introduced and commented in Section 2; assumed structure and stochastic enhancement of the model used for the evaluation of likelihood is introduced in Section 3; the experimental setup and results are reported in Section 4; limitations and further steps enabled by the results are discussed in Section 5.

2 Problem formulation

2.1 Description of the data set under consideration

We base our experimentation on a well-known and publicly available annotated data set for AR [27] containing binary data generated by 14 state-change sensors installed in a 3-room apartment, deployed at different locations (e.g., kitchen, bathroom, bedroom) and placed on various objects (e.g., household appliances, cupboards, doors). Seven activity types derived from the Katz Activities of Daily Living (ADL) index [14], i.e., $\Gamma = \{$*Leaving house*, *Preparing a beverage*, *Preparing breakfast*, *Preparing dinner*, *Sleeping*, *Taking shower*, *Toileting*$\}$, were performed and annotated by a 26-year-old subject during a period of 28 days. The annotation process yielded a *ground truth* consisting of a stream of activities a_1, $a_2, \ldots, a_K$, each being a triple $a_k = \langle \gamma_k, t_k^{start}, t_k^{end} \rangle$ where $\gamma_k \in \Gamma$ is the activity type, t_k^{start} is the activity start time, and t_k^{end} is the activity end time. An additional activity (not directly annotated) named *Idling* is considered, consisting of the time during which the subject is not performing any tagged activity. The data set includes 245 activity instances, plus 219 occurrences of *Idling*. Activities are usually annotated in a *mutually exclusive* way (i.e., one activity at a time), with the only exception of some instances of *Toileting* which was annotated so as to be performed concurrently with *Sleeping* (21 times) or with *Preparing dinner* (1 time).

The data set includes 1319 sensor events, classified in 14 event types, and encoded in the so called *raw* representation, which holds a high signal in the interval during which the condition detected by a sensor is true, and low otherwise (see Figure 1-left). In this case, each event is a triple $e_n = \langle \sigma_n, t_n^{start}, t_n^{end} \rangle$ where $\sigma_n \in \Sigma^{raw}$ is the event type, t_n^{start} is the event start time, and t_n^{end} is the event end time. As suggested in [2] for the handling of data sets with frequent object interaction, raw events where converted into a *dual change-point* representation, which emits a punctual signal when the condition goes true and when it goes back false (see Fig. 1-right). In this encoding, observations are a stream of punctual events $e_1, e_2, \ldots, e_N$ (doubled in number w.r.t. the raw representation, and sub-typed as *start_* and *end_*), each represented as a pair $e_n = \langle \sigma_n, t_n \rangle$, where $\sigma_n \in \Sigma$ is the event type, and t_n is the event occurrence time. In so doing, the number of events and event types has doubled, i.e., $N = 2638$, and $|\Sigma| = 28$.

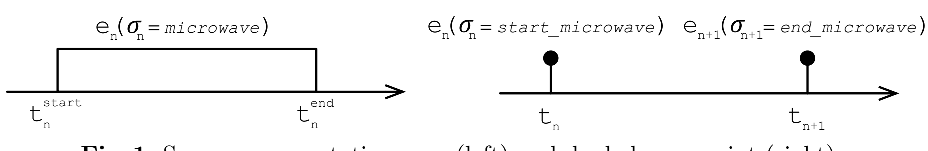

Fig. 1. Sensor representation: raw (left) and dual change-point (right).

Also for the limited accuracy of the tagging process (in [27], annotation was performed on-the-fly by tagging the start time t_k^{start} and the end time t_k^{end} of each performed activity a_k using a bluetooth headset combined with speech recognition software), the starting and ending points of activities are often delayed and anticipated, respectively. As a result, as shown in Figure 2, the start (end) time of an activity does not necessarily coincide with the occurrence time of its first (last) event.

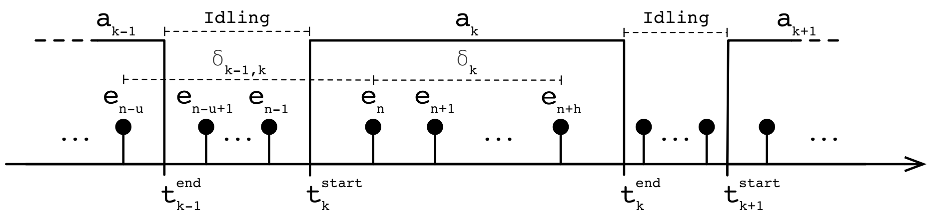

Fig. 2. A fragment of an events stream together with annotated activities.

2.2 Statistical abstraction

We abstracted the data set content so as to capture four aspects of its statistics: the duration of each activity; the time elapsed between subsequent events within each activity; the type of events occurred in each activity; and, the type of events occurred as first events in each activity.

Let $\{e_n = \langle \sigma_n, t_n \rangle\}_{n=1}^{N}$ and $\{a_k = \langle \gamma_k, t_k^{start}, t_k^{end} \rangle\}_{k=1}^{K}$ be the streams of observed events and tagged activities, respectively. The duration δ_k of an activity instance a_k is computed as the time elapsed from the first to the last event observed during a_k, i.e., $\delta_k = \max_{n|t_k^{start} \leq t_n \leq t_k^{end}} \{t_n\} - \min_{n|t_k^{start} \leq t_n \leq t_k^{end}} \{t_n\}$ (e.g., in Fig. 2, $\delta_k = t_{n+h} - t_n$). The duration $\delta_{k-1,k}$ of an instance of *Idling* enclosed within two activities a_{k-1} and a_k is derived as the time elapsed from the last event observed during a_{k-1} to the first event observed during a_k, i.e., $\delta_{k-1,k} = \min_{n|t_k^{start} \leq t_n \leq t_k^{end}} \{t_n\} - \max_{n|t_{k-1}^{start} \leq t_n \leq t_{k-1}^{end}} \{t_n\}$ (e.g., in Fig. 2, $\delta_{k-1,k} = t_n - t_{n-u}$). The *duration statistic* provides the mean and variation coefficient (CV) of the duration of each activity type, as shown in Table 1.

The *inter-events time statistic* evaluates the time between consecutive events occurring within an activity. In so doing, we do not distinguish between event types, and we only consider times between events. The inter-events time of *Idling* is computed taking into account *orphan events*, i.e., events not belonging to any tagged activity (e.g., e_{n-1} in Fig. 2), and the first event of each activity (e.g., e_n in Fig. 2). Also the inter-events time statistic provides mean and CV for each activity type, as shown in Table 1. Most of measured time spans have a CV higher than 1, thus exhibiting a hyper-exponential trend, as expected in ADL where timings may follow different patterns from time to time. Only the duration of *Leaving house* has a CV nearly equal to 1.

Table 2 shows the *event type statistic*, which computes the frequency $\psi_{\sigma,\gamma}$ of an event of type σ within an activity of type γ, $\forall\ \sigma \in \Sigma$, $\forall\ \gamma \in \Gamma$, i.e., $\psi_{\sigma,\gamma} = Prob\{$the type of the next event is σ | the type of the current activity is $\gamma\}$.

Table 3 shows the *starting event type statistic*, which evaluates: *i*) the frequency θ_σ of an event type σ either as the first event of an activity (regardless

	Duration		Inter-events time	
	μ (s)	**CV**	μ (s)	**CV**
Idling	1793.102	2.039	3906.167	3.292
Leaving house	40261.455	1.042	9354.190	2.810
Preparing a beverage	35.667	1.361	7.643	2.613
Preparing breakfast	108.684	0.713	9.928	1.844
Preparing dinner	1801.889	0.640	77.966	2.589
Sleeping	26116.571	0.442	1871.836	3.090
Taking shower	485.910	0.298	102.788	1.969
Toileting	88.742	1.175	14.814	2.449

Table 1. Activity duration and inter-events time statistics (trained on all days but the first one): mean (μ) and coefficient of variation (CV) of the duration and inter-events time of each activity, respectively.

of the activity type) or as an orphan event, $\forall\ \sigma \in \Sigma$, i.e., $\theta_\sigma = Prob\{$the type of an event e is $\sigma \mid e$ is either the first event observed during the current activity or an orphan event$\}$; and, *ii*) the frequency $\phi_{\sigma,\gamma}$ of an event of type σ as the first event of an activity of type γ, $\forall\ \sigma \in \Sigma$, $\forall\ \gamma \in \Gamma$, i.e., $\phi_{\sigma,\gamma} = Prob\{$an activity of type γ is started $\mid$ an event of type σ is observed and the subject was idling before the observation$\}$. As a by-product, $\forall\ \sigma \in \Sigma$, the statistic also computes $Prob\{$the subject remains idling $\mid$ an event of type σ is observed and the subject was idling before the observation$\} = 1 - \sum_{\gamma \in \Gamma} \phi_{\sigma,\gamma}$.

3 Classification Technique

In the proposed approach, a continuous-time stochastic model is constructed so as to fit the statistical characterization of the data set. Activity recognition is then based on a measure of likelihood that depends on the probability that observed time-stamped events have in this model.

3.1 Model syntax and semantics

The stochastic model is specified as a stochastic Time Petri Net (sTPN) [28]. As in [23], the formalism is enriched with flush functions which permit the marking of a set of places be reset to zero upon firing of a transition. This improves modeling convenience without any substantial impact on the complexity for the analysis.

Syntax An sTPN is a tuple $\langle P; T; A^-; A^+; A^\cdot; m_0;\ EFT; LFT; \mathcal{F}; \mathcal{C}; L\rangle$ where: P is the set of places; T is the set of transitions; $A^- \subseteq P \times T$, $A^+ \subseteq T \times P$, and $A^\cdot \subseteq P \times T$ are the sets of precondition, postcondition, and inhibitor arcs, respectively; $m_0 : P \to \mathbb{N}$ is the initial marking associating each place with a number of tokens; $EFT : T \to \mathbb{Q}_0^+$ and $LFT : T \to \mathbb{Q}_0^+ \cup \{\infty\}$ associate each transition with an *earliest* and a *latest firing time*, respectively, such that

	Leaving house	Preparing a beverage	Preparing breakfast	Preparing dinner	Sleeping	Taking shower	Toileting
start_front door	0.497	-	-	-	-	-	-
end_front door	0.503	-	-	-	-	-	-
start_hall-bathroom door	-	-	-	0.018	0.111	0.008	0.261
end_hall-bathroom door	-	-	-	0.023	0.115	-	0.261
start_hall-bedroom door	-	-	-	-	0.274	-	0.019
end_hall-bedroom door	-	-	-	-	0.280	-	0.013
start_hall-toilet door	-	-	0.004	-	0.041	0.540	0.057
end_hall-toilet door	-	-	0.004	-	0.045	0.452	0.063
start_cups cupboard	-	0.176	0.009	0.018	-	-	-
end_cups cupboard	-	0.176	0.009	0.018	-	-	-
start_groceries cupboard	-	-	0.119	0.055	-	-	-
end_groceries cupboard	-	-	0.123	0.055	-	-	-
start_pans cupboard	-	-	0.009	0.115	-	-	-
end_pans cupboard	-	-	0.009	0.111	-	-	-
start_plates cupboard	-	-	0.106	0.083	-	-	-
end_plates cupboard	-	-	0.106	0.083	-	-	-
start_dishwasher	-	0.010	0.004	0.005	-	-	-
end_dishwasher	-	0.010	0.004	0.005	-	-	-
start_freezer	-	0.020	0.049	0.070	-	-	-
end_freezer	-	0.020	0.049	0.070	-	-	-
start_fridge	-	0.294	0.167	0.106	-	-	-
end_fridge	-	0.294	0.167	0.106	-	-	-
start_microwave	-	-	0.031	0.023	-	-	-
end_microwave	-	-	0.031	0.018	-	-	-
start_toilet flush	-	-	-	0.009	0.067	-	0.163
end_toilet flush	-	-	-	0.009	0.067	-	0.163

Table 2. Event type statistic (trained on all days but the first one): frequency $\psi_{\sigma,\gamma}$ of each event type $\sigma \in \Sigma$ (rows) within each activity type $\gamma \in \Gamma$ (columns).

$EFT(t) \leq LFT(t)\ \forall\ t \in T$; $\mathcal{F} : T \to F_t^s$ associates each transition with a static Cumulative Distribution Function (CDF) with support $[EFT(t), LFT(t)]$; $\mathcal{C} : T \to \mathbb{R}^+$ associates each transition with a weight; $L : T \to \mathcal{P}(P)$ is a a *flush function* associating each transition with a subset of P. A place p is termed an *input*, an *output*, or an *inhibitor* place for a transition t if $\langle p, t\rangle \in A^-$, $\langle t, p\rangle \in A^+$, or $\langle p, t\rangle \in A^{\cdot}$, respectively. A transition t is called *immediate* (IMM) if $[EFT(t), LFT(t)] = [0, 0]$ and *timed* otherwise; a timed transition t is termed *exponential* (EXP) if $F_t(x) = 1 - e^{-\lambda x}$ over $[0, \infty]$ for some rate $\lambda \in \mathbb{R}_0^+$ and *general* (GEN) otherwise; A GEN transition t is called *deterministic* (DET) if $EFT(t) = LFT(t)$ and *distributed* otherwise. For each distributed transition t,

	frequency θ_σ	Leaving house	Preparing a beverage	Preparing breakfast	Preparing dinner	Sleeping	Taking shower	Toileting	Idling
start_front door	0.076	0.780	-	-	-	-	-	-	0.220
end_front door	0.017	0.111	-	-	-	-	-	-	0.889
start_hall-bathroom door	0.183	-	-	-	-	-	-	0.707	0.293
end_hall-bathroom door	0.063	-	-	-	-	-	-	0.118	0.882
start_hall-bedroom door	0.050	-	-	-	-	0.148	-	0.296	0.556
end_hall-bedroom door	0.046	-	-	-	-	0.360	-	0.080	0.560
start_hall-toilet door	0.117	-	-	-	-	0.095	0.222	0.016	0.667
end_hall-toilet door	0.128	-	-	0.015	-	0.029	0.116	0.145	0.695
start_cups cupboard	0.048	-	0.308	-	-	-	-	-	0.692
start_groceries cupboard	0.044	-	-	-	0.125	-	-	-	0.875
start_pans cupboard	0.031	-	-	0.118	-	-	-	-	0.882
start_plates cupboard	0.046	-	-	0.520	-	-	-	-	0.480
start_dishwasher	0.024	-	-	-	0.077	-	-	-	0.923
start_freezer	0.020	-	0.091	-	0.273	-	-	-	0.636
start_fridge	0.068	-	0.243	0.081	0.054	-	-	-	0.622
start_toilet flush	0.039	-	-	-	-	-	-	0.524	0.476

Table 3. Starting event type statistic (trained on all days but the first one): for each event type $\sigma \in \Sigma$, *i)* frequency θ_σ (first column), *ii)* frequency $\phi_{\sigma,\gamma}$ for each activity type $\gamma \in \Gamma$ (from the second to the second-to-last column), *iii)* $1 - \sum_{\gamma \in \Gamma} \phi_{\sigma,\gamma}$ (the last column). Note that only the 16/28 event types that have non-null θ_σ are shown.

we assume that F_t is absolutely continuous over its support and thus that there exists a Probability Density Function (PDF) f_t such that $F_t(x) = \int_0^x f_t(y)dy$.

Semantics The *state* of an sTPN is a pair $\langle m, \tau \rangle$, where $m : P \rightarrow \mathbb{N}$ is a marking and $\tau : T \rightarrow \mathbb{R}_0^+$ associates each transition with a time-to-fire. A transition is *enabled* if each of its input places contains at least one token and none of its inhibitor places contains any token; an enabled transition is *firable* if its time-to-fire is not higher than that of any other enabled transition. When multiple transitions are firable, one of them is selected to fire with probability $Prob\{t \text{ is selected}\} = \mathcal{C}(t)/\sum_{t_i \in T^f(s)} \mathcal{C}(t_i)$, where $T^f(s)$ is the set of firable transitions in s. When t fires, $s = \langle m, \tau \rangle$ is replaced by $s' = \langle m', \tau' \rangle$, where m' is derived from m by: *i)* removing a token from each input place of t and assigning zero tokens to the places in $L(t) \subseteq P$, which yields an intermediate marking m_{tmp}, *ii)* adding a token to each output place of t. Transitions enabled both by m_{tmp} and by m' are said *persistent*, while those enabled by m' but not by m_{tmp} or m are said *newly-enabled*; if t is still enabled after its own firing, it is regarded as newly enabled [4]. The time-to-fire of persistent transitions is reduced by the time elapsed in s, while the time-to-fire of newly-enabled transitions takes a random value sampled according to their CDF.

3.2 Model structure and enhancement

Model structure The model used to evaluate the likelihood of observed events is organized by composition of 7+1 submodels, which fit the observed behavior in the 7 activities classified in the Van Kasteren data set [27] and in the remaining *Idling* periods. Fig. 3 shows a fragment focused on *Idling* and *Preparing a beverage*.

In the *Idling* submodel, places `p_IDLE_durationStart` and `p_IDLE_eventWait` receive a token each when the *Idling* starts, on completion of any activity. The token in `p_IDLE_eventWait` is removed whenever an event is observed (firing of the GEN transition `t_IDLE_interEventsTime`); in this case, different IMM transitions are fired depending on the type of the observed event (`t_START_FRIDGE`, `t_START_FREEZER`, ...), and then for each different type, a choice is made on whether the event is interpreted as a continuation of the *Idling* period (e.g., `t_skip_START_FRIDGE`) which will restore a token in `p_IDLE_eventWait`, or as the starting of each of the possible activities (e.g. `t_START_FRIDGE_starts_GET_DRINK`, `t_START_FRIDGE_starts_PREPARE_DINNER`,...). In parallel to all this, the token in `p_IDLE_durationStart` will be removed when the duration of *Idling* expires (at the firing of the GEN transition `t_IDLE_duration`) or when an observed event is interpreted as the beginning of any activity (e.g., at the firing of `t_START_FRIDGE_starts_GET_DRINK`); note that the latter case is not shown in the graphical representation and it is rather encoded in a flush function. Similarly, when the duration of *Idling* expires, the token in `p_IDLE_eventWait` will be removed by a flush function associated with transition `t_GET_IDLE_duration`.

In the *Preparing a beverage* submodel, places `p_GET_DRINK_durationStart` and `p_GET_DRINK_eventWait` receive a token each when an event observed during the *Idling* period is interpreted as the beginning of an instance of the *Preparing a beverage* activity. The token in `p_GET_DRINK_eventWait` is removed whenever an event is observed (firing of the GEN transition `t_GET_DRINK_interEventsTime`), and restored after the event is classified according to its type (IMM transitions `t_GET_DRINK_END_FRIDGE`, `t_GET_DRINK_END_FREEZER`, ...). In parallel to this, the token in `p_GET_DRINK_durationStart` will be removed when the duration of the *Preparing a beverage* activity expires (at the firing of the GEN transition `t_GET_DRINK_duration`). Note that, in this case, also the token in `p_GET_DRINK_eventWait` will be removed: this is performed by a flush function associated with transition `t_GET_DRINK_duration`.

In so doing, in any reachable marking, only the submodel of the current activity contains two non-empty places, one indicating that the activity duration is elapsing (e.g., `p_GET_DRINK_durationStart`) and the other one meaning that the inter-events time is expiring (e.g., `p_GET_DRINK_eventWait`). Note that, as a naming convention, any transition named `t_EVENT` (where `EVENT` is an event type that may start an activity) or `t_ACTIVITY_EVENT` (where `EVENT` is an event type that may occur within ACTIVITY) accounts for an *observable* event, while all the other transitions correspond to *unobservable* events. Finally, note that the general structure of the model is open to modifications in various directions. For instance, in the submodels of activities, the choice between events might be easily made dependent on the duration of the inter-event time, which would allow a more precise classification without significantly impacting on the analyzability of the model. The viability of such evolutions mainly depends on the significance of the statistics that can be derived form the data set.

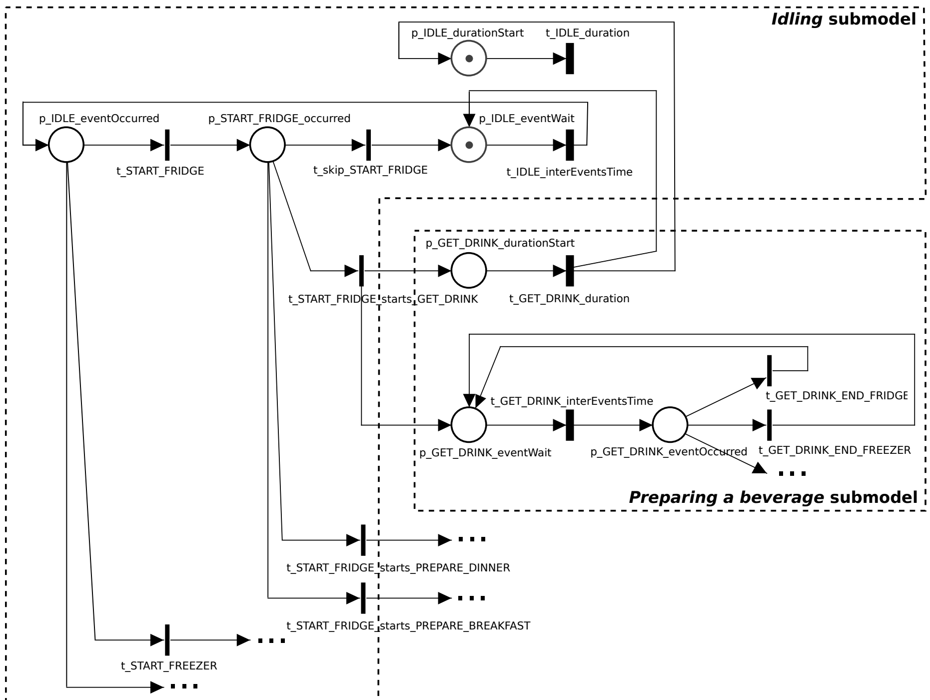

Fig. 3. A fragment of the stochastic model used to evaluate the likelihood measure.

Model enhancement The actual topology of the model and its stochastic temporal parameters are derived in automated manner from the statistical indexes extracted in the abstraction of the data set (Section 2).

The event types that can start an activity (e.g., in the model of Fig. 3, `t_START_FRIDGE_starts_GET_DRINK`, `t_START_FRIDGE_starts_PREPARE_DINNER`, ...) and the discrete probabilities in their random switch are derived from the *starting event type* statistic. The event types that can be observed during each activity (e.g., `t_GET_DRINK_END_FRIDGE`, `t_GET_DRINK_END_FREEZER`, ...) or can continue an *Idling* period and the discrete probabilities in their random switch are derived from the *event type* statistics.

The distribution associated with GEN transitions is derived from the *duration* statistic and the *inter-events time* statistic by fitting expected value and

Coefficient of Variation (CV) as in [29]: if $0 \leq CV \leq 1/\sqrt{2}$, we assume a shifted exponential distribution with PDF $f(x) = \lambda e^{-\lambda(x-d)}$ over $[d, \infty)$, where σ^2 is the variance, $\lambda = \sigma^{-1}$, and $d = \mu - \sigma$; if $1/\sqrt{2} < CV < 1$, we use a hypo-exponential distribution with PDF $f(x) = \lambda_1\lambda_2/(\lambda_1 - \lambda_2)(e^{-x\lambda_2} - e^{-x\lambda_1})$, where $\lambda_i^{-1} = (\mu/2)(1 \pm \sqrt{2CV^2 - 1})$, with $i = 1, 2$; if $CV \approx 1$, we adopt an exponential distribution with $\lambda = 1/\mu$; if $CV > 1$, we consider a hyper-exponential distribution with PDF $f(x) = \sum_{i=1}^{2} p_i \lambda_i e^{-\lambda_i x}$, where $p_i = [1 \pm \sqrt{(CV^2 - 1)/(CV^2 + 1)}]/2$ and $\lambda_i = 2p_i\mu^{-1}$.

3.3 Online diagnosis and prediction

We use the transient probability $\pi_\gamma(t)$ that an activity of type $\gamma \in \Gamma$ is being performed at time t as a *measure of likelihood* for γ at t. A prediction $\mathcal{P}(t)$ emitted at time t consists of the set of activity types that may be performed at t, each associated with the likelihood measure, i.e., $\mathcal{P}(t) = \{\langle \gamma, \pi_\gamma(t) \rangle \mid \pi_\gamma(t) \neq 0\}$. Between any two subsequent events $e_n = \langle \sigma_n, t_n \rangle$ and $e_{n+1} = \langle \sigma_{n+1}, t_{n+1} \rangle$, a prediction $\mathcal{P}(t)$ is emitted at equidistant time points in the interval $[t_n, t_{n+1}]$, i.e., $\forall\ t \in \{t_n,\ t_n + q,\ t_n + 2q,\ \ldots,\ t_{n+1}\}$, with $q \in \mathbb{R}^+$. In the experiments, we assume the activity type with the highest measure of likelihood as the predicted class to be compared against the actual class annotated in the ground truth, i.e., at time t, the predicted activity is $\gamma \mid \pi_\gamma(t) = \max_{a \in \Gamma \mid \langle a, \pi_a(t) \rangle \in \mathcal{P}(t)} \{\pi_a(t)\}$.

As a result of the prescribed model structure and the specific enhancement, the stochastic model subtends a Markov Regenerative Process (MRP) [9, 10, 5] under enabling restriction, i.e., no more than one GEN transition is enabled in each marking (only the duration of four activities is modeled by a shifted exponential distribution, thus no more than one DET transition is enabled in each marking). According to this, online diagnosis and prediction can be performed by leveraging the regenerative transient analysis of [12]. The solution technique of [12] samples the MRP state after each transition firing, maintaining an additional timer τ_{age} accounting for the absolute elapsed time; each sample, called *transient class*, is made of the marking and the joint PDF of τ_{age} and the times-to-fire of the enabled transitions. Within a given time limit T, enumeration of transient classes is limited to the first regeneration epoch and repeated from every regeneration point (i.e., a state where the future behavior is independent from the past), enabling the evaluation of transient probabilities of reachable markings through the solution of generalized Markov renewal equations.

In the initial transient class of the model, the marking assigns a token to places `p_IDLE_durationStart` and `p_IDLE_eventWait`, all transitions are newly enabled, and τ_{age} has a deterministic value equal to zero. After n observed events $e_1 = \langle \sigma_1, t_1 \rangle,\ \ldots,\ e_n = \langle \sigma_n, t_n \rangle$, let $S_n = \{\langle s_n^i, \omega_n^i \rangle\}$ be the set of possible transient classes s_n^i having probability ω_n^i, where $\sum_{i \mid \langle s_n^i, \omega_n^i \rangle \in S_n} \omega_n^i = 1$. Regenerative transient analysis [12] of the model is performed from each possible transient class $\langle s_n^i, \omega_n^i \rangle \in S_n$ up to any observable event within a given time limit, which is set equal to 48 h to upper bound the time between any two subsequent events. This allows one to evaluate transient probabilities of reachable markings, i.e., $p_m^{n,i}(t) = Prob\{M^{n,i}(t) = m\}\ \forall\ t \leq T,\ \forall\ m \in \mathcal{M}^{n,i}$, where

$\mathbb{M}^{n,i} = \{M^{n,i}(t), t \geq 0\}$ is the underlying marking process, and $\mathcal{M}^{n,i}$ is the set of markings that are reachable from s_n^i. Since in any reachable marking only the submodel of the ongoing activity contains non-empty places, transient probabilities of markings are aggregated to derive transient probabilities of ongoing activities $\pi_\gamma(t) = \sum_{\langle s_n^i, \omega_n^i\rangle \in S_n} \omega_n^i \sum_{m \in \mathcal{M}_\gamma^{n,i}} p_m^{n,i}(t) \; \forall \; \gamma \in \mathit{\Gamma}$, where $\mathcal{M}_\gamma^{n,i}$ is the set of markings that are reachable from s_n^i and have non-empty places in the submodel of the activity type γ.

Whenever an event $e_{n+1} = \langle \sigma_{n+1}, t_{n+1}\rangle$ is observed, any tree of transient classes enumerated from class $\langle s_n^i, \omega_n^i\rangle \in S^n$ is explored to determine the possible current classes and their probability. More specifically: *i*) the possible current classes are identified as those classes that can be reached after a time $t_{n+1} - t_n$ through a sequence of unobservable events followed by the observed event e_{n+1}; *ii*) any possible current class s_{n+1}^j is a regeneration point since, by model construction, each GEN transition is either newly enabled or enabled by a deterministic time (i.e., the timestamp $t_{n+1} - t_n$); and, *iii*) the probability ω_{n+1}^j of s_{n+1}^j is obtained as $\lim_{t \to t_{n+1}^-} \zeta_{s_{n+1}^p}(t) \cdot \rho$, where s_{n+1}^p is the last class where the model waits for the arrival of the next event e_{n+1}, $\zeta_{s_{n+1}^p}(t)$ is the probability of being in class s_{n+1}^p at time t, and ρ is the product of transition probabilities of the arcs encountered from s_{n+1}^p to s_{n+1}^j; *iv*) in the limit case that s_{n+1}^p is vanishing, ω_{n+1}^j is obtained as the product of transition probabilities of the arcs encountered from the root class to s_{n+1}^j. Hence, the approach is iterated, performing transient analysis from any new current class up to any observable event, still encountering regeneration points after each observed event.

By construction, the approach complexity is linear in the number of observed events. For each observed event $e_n = \langle \sigma_n, t_n\rangle$, the number of transient trees to enumerate is proportional to the number of possible parallel hypotheses $|\mathcal{P}(t_n)|$, i.e., the number of activity types that may be performed at time t_n; moreover, the depth of each transient tree is proportional to the number of events that may occur between two observations (which is bounded in the considered application context), and relatively insensitive to the time elapsed between observed events.

4 Computational Experience

4.1 Experimental setup

Experiments were performed on the data set [27], using a *dual change-point* representation for sensor events as detailed in Section 2. We split data provided by the computed statistics and event logs into training and test sets using a *Leave One Day Out* (LOO) approach, which consists of using each instance of one full day sensor readings for testing and the instances of the remaining days for training. Since, in each test, predictions are emitted starting from the first observed event of the day, we extended online analysis up to the first event of the next day. To avoid inconsistencies in the characterization of *Leaving house*, during which all the event types were observed, we removed from the training sets all events occurring during *Leaving house* that are not of type *start_front door* and *end_front door*. Moreover, whenever the ground truth includes concurrent activities, we considered our prediction correct if the predicted activity type is equal to any of the concurrent activity types.

We performed experiments using two fitting techniques in the evaluation of the duration and the inter-events time statistics: *i*) only exponential distributions (i.e., *exponential* case); *ii*) different classes of distributions based on the CV value, as discussed in Section 3.2 (i.e., *non-Markovian* case). On a machine with an Intel Xeon 2.67 GHz and 32 GB RAM, the evaluation for a single day took on average 43 s for the exponential case and 18 minutes for the non-Markovian case.

We evaluated the performance of our approach computing, for each activity class, three measures, derived from the number of true positives (TP), false positives (FP), and false negatives (FN): *i*) *precision* $Pr = TP/(TP + FP)$, which accounts for the accuracy provided that a specific class has been predicted; *ii*) *recall* $Re = TP/(TP + FN)$, which represents the ability to select instances of a certain class from a data set; and, *iii*) *F-measure* $F_1 = 2*Pr*Re/(Pr+Re)$, which is the harmonic mean of precision and recall.

We also compared the outcome of our experiments with the results reported in [27], obtained using a generative model (i.e., an HMM) and a discriminative one (i.e., a CRF) in combination with *offline* inference and the change-point representation. To make this comparison possible, we sampled the result of our prediction using a timeslice of duration $\Delta t = 60\ s$ and we considered two additional measures: *i*) *accuracy* $A = 1/N \sum_{i=1}^{N_c} TP_i$, which is the average percentage of correctly classified timeslices, with N being the total number of timeslices, N_c the total number of activity classes, and TP_i the number of TP of class i; and, *ii*) *average recall* $\bar{Re} = 1/N_c \sum_{i=1}^{N_c} Re_i$, which is the average percentage of correctly classified timeslices per class, with Re_i being the recall of class i.

4.2 Results

Table 4 shows the confusion matrix for the exponential and the non-Markovian cases, which reports in position i, j the number of timeslices of class i predicted as class j; Table 5 shows precision, recall and F_1 score as derived from the confusion matrix. Results show that *Idling*, *Leaving house*, and *Sleeping* are the activities with the highest F_1 score. In terms of F_1 score, the non-Markovian case outperforms the exponential one for all activity classes except for *Preparing a beverage* and *Taking shower*. In terms of precision, the non-Markovian case performs worse only for *Sleeping*. Conversely, in terms of recall, the exponential case outperforms the non-Markovian one for *Preparing a beverage*, *Preparing breakfast*, *Taking shower*, and *Toileting*, and performs worse for *Idling*, *Preparing dinner*, and *Sleeping*. Note that the precision, recall, and F_1 score of *Leaving house* are identical in both cases.

Accuracy and average recall are summarized in Table 6, and compared with results from [27]. As we can see, fitting statistical data according to the CV (non-Markovian case), we achieve the highest accuracy, both w.r.t. our exponential case and w.r.t. HMM and CRF. Nevertheless, the exponential case, HMM, and CRF outperform the non-Markovian case in terms of average recall.

	Idling	Leaving house	Preparing a beverage	Preparing breakfast	Preparing dinner	Sleeping	Taking shower	Toileting
Idling	2975/3471	330/330	37/6	82/16	733/400	588/658	131/65	98/28
Leaving house	184/209	22219/22219	1/0	0/0	101/56	25/71	22/4	15/8
Preparing a beverage	8/6	1/1	5/2	0/0	6/11	1/1	0/0	0/0
Preparing breakfast	5/14	0/0	3/2	39/24	15/22	8/8	0/0	0/0
Preparing dinner	83/107	0/0	4/1	39/13	214/221	0/0	0/0	1/0
Sleeping	194/215	0/0	0/0	0/0	0/0	11226/11430	3/1	231/8
Taking shower	94/100	0/0	0/0	0/0	1/1	17/41	105/79	4/0
Toileting	46/60	2/2	0/0	0/0	0/1	36/52	7/8	66/33

Table 4. Confusion matrix showing the number of timeslices of each class i (first column) predicted as class j (other columns): exponential case/non-Markovian case. Diagonal elements represent TP, while FN (FP) can be read along rows (columns).

	Exponential			**non-Markovian**		
	Precision	**Recall**	$\mathbf{F_1}$	**Precision**	**Recall**	$\mathbf{F_1}$
Idling	82.89	59.81	69.49	83.00	69.78	75.82
Leaving house	98.52	98.46	98.49	98.52	98.46	98.49
Preparing a beverage	10.00	23.81	14.09	18.18	9.52	12.50
Preparing breakfast	24.38	55.71	33.91	45.28	34.29	39.02
Preparing dinner	20.00	62.76	30.33	31.04	64.62	41.94
Sleeping	94.33	96.33	95.32	93.22	98.08	95.59
Taking shower	39.18	47.51	42.95	50.32	35.75	41.80
Toileting	15.90	42.04	23.08	42.86	21.15	28.33

Table 5. Precision, recall, and F_1 score achieved for each activity type.

	Accuracy	**Average recall**
Exponential case	92.11	60.80
non-Markovian case	93.69	53.96
HMM [27]	80.00	67.20
CRF [27]	89.40	61.70

Table 6. Accuracy and average recall achieved by the exponential and non-Markovian cases (dual change-point representation and online analysis), compared with those reported in [27] for HMM and CRF (change-point representation and offline analysis).

5 Discussion

Experimentation developed so far achieves results that compare well with the HMM and CRF approaches, with a slight increase in precision and a slight reduction in recall. The proposed approach is open to various possible developments, and the insight on observed cases of success and failure comprises a foundation for refinement and further research on which we are presently working.

In the present implementation, classification of the current activity relies only on past events, which is for us instrumental to open the way to the integration of classification with on-line prediction. However, the assumption of this causal constraint severely hinders our approach in the comparison against the offline classification implemented in [27] through HMM and CRF. Online classification results reported in [27] are unfortunately not comparable due to the different abstraction applied on events, and it should be remarked that in any case they are not completely online as the classification at time t relies on all the events that will be observed within the end of the timeslice that contains t itself, which makes a difference for short duration activities. For the purposes of comparison, our online approach can be relaxed to support offline classification by adding a backtrack from the final states reached by the predictor. This should in particular help the recall of short activities started by events that can be accepted also as the beginning of some longer activity (e.g., Preparing a beverage w.r.t. Preparing dinner). We also expect that this should reduce the number of cases where a time period is misclassified as Idling.

The statistics of durations are now fitted using the basic technique of [29] which preserves only expected value and variation coefficient. Moreover, the deterministic shift introduced in the approximation of hypo-exponential distributions with low CV causes a false negative for all the events occurring before the shift completion, which is in fact observed in various cases. Approximation through acyclic Continuous PHase type (CPH) distributions [17, 13, 21] would remove the problem and allow an adaptable trade-off between accuracy and complexity. In particular, a promising approach seems to be the method of [6] which permits direct derivation of an acyclic phase type distribution fitting not only expected value and variation coefficient but also skewness.

Following a different approach, the present implementation is completely open to the usage of any generally distributed (GEN) representation of activity durations. This would maintain the underlying stochastic process of the online model within the current class, i.e., Markov Regenerative Processes (MRP) that run under enabling restriction [10] and guarantee a regeneration within a bounded number of steps. In this case, on-line prediction could be practically implemented using various tools, including Oris [12, 7], TimeNet [31] and GreatSPN [1].

In the present implementation, classification is unaware of the absolute time, which may instead become crucial to separate similar activities, such as for instance Preparing breakfast and Preparing dinner. To overcome the limitation, the model should in principle become non-homogeneous, but a good approximation can be obtained by assuming a discretized partition of the daytime, which can be cast in the on-line model as a sequence of deterministic delays. By exploiting the timestamps, at each event the current estimation of the absolute time is restarted. Under the fair assumption that at least one event is observed in each activity, the underlying stochastic process of the on-line model still falls in the class of MRPs that encounter a regeneration within a bounded number of steps, and can thus be practically analyzed through the Oris Tool [12, 7].

References

1. E. G. Amparore, P. Buchholz, and S. Donatelli. A Structured Solution Approach for Markov Regenerative Processes. In *Int. Conference on Quantitative Evaluation of Systems*, pages 9–24. Springer, 2014.
2. U. Avci and A. Passerini. Improving Activity Recognition by Segmental Pattern Mining. *IEEE Trans. on Knowledge and Data Eng.*, 26(4):889–902, 2014.
3. Ezio Bartocci, Luca Bortolussi, and Guido Sanguinetti. Data-driven statistical learning of temporal logic properties. In *Proc. Int. Conf. Formal Modeling and Analysis of Timed Systems*, pages 23–37. Springer, 2014.
4. B. Berthomieu and M. Diaz. Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Trans. on Soft. Eng.*, 17(3):259–273, March 1991.
5. A. Bobbio and M. Telek. Markov regenerative SPN with non-overlapping activity cycles. *Int. Computer Performance and Dependability Symp.*, pages 124–133, 1995.
6. Andrea Bobbio, András Horváth, and Miklós Telek. Matching three moments with minimal acyclic phase type distributions. *Stochastic models*, 21(2-3):303–326, 2005.
7. G. Bucci, L. Carnevali, L. Ridi, and E. Vicario. Oris: a tool for modeling, verification and evaluation of real-time systems. *Int. Journal of SW Tools for Technology Transfer*, 12(5):391 – 403, 2010.
8. R. Buchholz, C. Krull, T. Strigl, and G. Horton. Using Hidden non-Markovian Models to Reconstruct System Behavior in Partially-Observable Systems. In *Int. ICST Conf. on Simulation Tools and Techniques*, page 86, 2010.
9. H. Choi, V. G. Kulkarni, and K. S. Trivedi. Markov regenerative stochastic Petri nets. *Perf. Eval.*, 20(1-3):337–357, 1994.
10. G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Transactions on Software Engineering*, 20(7):506–515, 1994.
11. D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Comp.*, 5(4):277–298, 2009.
12. A. Horváth, M. Paolieri, L. Ridi, and E. Vicario. Transient analysis of non-Markovian models using stochastic state classes. *Performance Evaluation*, 69(7-8):315–335, 2012.
13. A. Horváth and M. Telek. Phfit: A general phase-type fitting tool. In *Int. Conf. Computer Performance Evaluation, Modelling Techniques and Tools - TOOLS 2002*, pages 82–91, 2002.
14. S. Katz, T. D. Downs, H. R. Cash, and Robert C Grotz. Progress in development of the index of adl. *The gerontologist*, 10(1 Part 1):20–30, 1970.
15. R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker. Application of Process Mining in Healthcare - A Case Study in a Dutch Hospital. In *Biomedical Eng. Sys. and Technologies*, pages 425–438. Springer, 2009.
16. C. D. Mitchell and L. H. Jamieson. Modeling duration in a hidden Markov model with the exponential family. In *Int. Conf. Acoustics, Speech, and Signal Processing*, volume 2, pages 331–334. IEEE, 1993.

17. M. F. Neuts. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, 1981.
18. D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *Ubiquitous Computing*, pages 73–89, 2003.
19. P. Rashidi and D. J. Cook. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(5):949–959, 2009.
20. P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe. Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering*, 23(4):527–539, 2011.
21. P. Reinecke, T. Krauß, and K. Wolter. Phase-type fitting using hyperstar. In *Europ. Workshop on Computer Perf. Eng.*, pages 164–175, 2013.
22. A. Rogge-Solti and M. Weske. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays. In *Service-Oriented Computing*, pages 389–403. Springer, 2013.
23. K. S. Trivedi. *Probability and statistics with reliability, queuing, and computer science applications.* John Wiley and Sons, New York, 2001.
24. W. Van Der Aalst, A. Adriansyah, A. K. Alves de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, et al. Process mining manifesto. In *Business process management workshops*, pages 169–194. Springer, 2012.
25. W. M. P. van der Aalst, H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. Alves De Medeiros, M. Song, and H. M. W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, 2007.
26. B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. Van Der Aalst. The ProM framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005*, pages 444–454. Springer, 2005.
27. T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proc. Int. Conf. on Ubiquitous Computing*, UbiComp '08, pages 1–9, New York, NY, USA, 2008. ACM.
28. E. Vicario, L. Sassoli, and L. Carnevali. Using stochastic state classes in quantitative evaluation of dense-time reactive systems. *IEEE Transactions on Software Engineering*, 35(5):703–719, 2009.
29. W. Whitt. Approximating a point process by a renewal process, i: Two basic methods. *Operations Research*, 30(1):125–147, Jan.-Feb. 1982.
30. J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and mobile computing*, 8(1):36–66, 2012.
31. A. Zimmermann. Dependability Evaluation of Complex Systems With TimeNET. In *Proc. Int. Workshop on Dynamic Aspects in Dependability Models for Fault-Tolerant Systems*, pages 33–34, 2010.