UDC 004.93
DOI: 10.31891/CSIT-2021-5-16

VITALIY YAKOVYNA, BOHDAN UHRYNOVSKYI
Lviv Polytechnic National University, Lviv, Ukraine

EXTENDED SOFTWARE AGING AND REJUVENATION MODEL FOR ANDROID OPERATING SYSTEM CONSIDERING DIFFERENT AGING LEVELS AND REJUVENATION PROCEDURE TYPES

A complex model based on Continuous-Time Markov Chains is proposed, which combines an extended aging and rejuvenation model taking into account different aging levels and a model of mobile device usage activity. A graph of states and transitions is constructed, which describes the proposed model without taking into account mobile device usage activity, and taking it into account. A system of Kolmogorov – Chapman differential equations is written on the basis of the states graph. A set of test simulations for conducting experimental calculations of the model and analysis of results is described. A system of differential equations for each simulation is calculated using the 4th order Runge-Kutta method. The analysis of simulations with recovery after aging-related failure and without recovery allowed to formulate the main objectives of the rejuvenation procedure in the proposed model to improve the user experience. Analysis of different rejuvenation planning strategies indicates that the most effective approach is to perform rejuvenation in the "Aging" state, when the device is already aging, but it is not yet a state with a high probability of aging-related failure. Analysis of simulations with warm and cold rejuvenation shows that this factor affects the results of the model calculation, and the application of one or another approach depends on the aging conditions and the mobile device usage activity. The developed model based on the Markov chain can be used to predict the optimal time of the rejuvenation procedure. In addition, the model considers both cold and warm rejuvenation. Further studies which take into account the real data and aging conditions are needed for proposed aging and rejuvenation model.

Keywords: software aging, software rejuvenation, Markov Chains, Android.

ВІТАЛІЙ ЯКОВИНА, БОГДАН УГРИНОВСЬКИЙ
Національний університет «Львівська політехніка»

МОДЕЛЬ ПРОЦЕСУ СТАРІННЯ ТА ОМОЛОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID З УРАХУВАННЯМ РІЗНИХ РІВНІВ СТАРІННЯ ТА ТИПІВ ПРОЦЕДУРИ ОМОЛОДЖЕННЯ

Явище старіння програмного забезпечення негативно впливає на його надійність і продуктивність, що позначається на погіршенні користувацького досвіду. В свою чергу, мобільні пристрої є особливо вразливі до явища старіння та вимагають високого рівня якості програмного забезпечення, тому важливим завданням є розроблення засобів для протидії старінню в мобільних пристроях.

В даній роботі запропоновано комплексну модель на основі ланцюга Маркова з неперервним часом, яка об'єднує розширену модель старіння і омолодження програмного забезпечення з урахуванням різних рівнів старіння та модель активності використання мобільного пристрою користувачем. Побудовано граф станів та переходів, що описує розширену модель старіння та омолодження без урахування моделі активності використання мобільного пристрою користувачем, та з її урахуванням. На основі графу станів написано систему диференційних рівнянь Колмогорова–Чепмена. Описано набір тестових симуляцій для проведення експериментальних обчислень моделі та аналізу результатів. Обчислено систему диференційних рівнянь для кожної симуляції з допомогою методу Рунге-Кутти 4-го порядку.

Аналіз симуляцій із відновленням після відмови старіння та без відновлення дозволив сформулювати основні завдання процедури омолодження в запропонованій моделі для покращення користувацького досвіду. Аналіз різних стратегій планування омолодження вказує, що найефективнішим підходом є планування омолодження в стані «Aging», коли в пристрої вже спостерігається старіння, але це ще не стан з високою імовірністю відмови старіння. Аналіз симуляцій із «warm» та «cold» омолодженням показує, що цей чинник впливає на результати обчислення моделі, а застосування того чи іншого підходу залежить від умов старіння та активності використання мобільного пристрою.

Розроблена модель на основі ланцюга Маркова може бути застосована для прогнозування оптимального часу виконання процедури омолодження. Крім того, модель дозволяє враховувати як cold омолодження, так і warm омолодження. Необхідні подальші дослідження розробленої моделі старіння та омолодження для реальних даних і умов старіння.

Ключові слова: старіння програмного забезпечення, омолодження програмного забезпечення, ланцюг Маркова, операційна система Android.

Introduction

Performance and reliability are important characteristics of software quality [1]. The software aging phenomenon [2, 3] has a negative effect on software reliability and performance. Software aging is the process of software response time and aging-related failure [4] rate increasing, which is a consequence of the aging-related errors accumulation [4] in systems that run for a long time.

Software aging phenomenon is observed in many systems [4-6]. The widespread use of mobile devices in everyday life requires a high level of software and mobile systems quality, in particular, Android operating system (OS) [7, 8]. It is also worth noting that mobile devices have a number of characteristics [9] that distinguish them from personal computers and servers, in particular, the usage activity of mobile device, as well as limited hardware resources, such as random-access memory (RAM) or central processing unit (CPU).

To counteract the negative effects of software aging, a proactive approach is used, namely, software rejuvenation [3, 5, 10]. The main task of software rejuvenation is to clean the accumulated aging-related errors by rebooting the system or its individual components, which ensures that the system is constantly in a highly productive state and prevents the occurrence of aging-related failures. In turn, the special features of mobile systems should be taken into account when developing software aging and rejuvenation (SAR) models and methods for mobile systems.

Performing a rejuvenation procedure can improve performance and reliability, however, it also may have a negative impact on the user experience. For example, frequent rejuvenation by rebooting the entire OS or early rejuvenation when no aging is observed can lead to an unreasonable increase of system downtime. Late rejuvenation, when the system is already likely to fail due to aging-related error, is also ineffective, as there is a risk that the planned rejuvenation will not occur. To address the described features of choosing the optimal time and method of mobile device rejuvenation, in particular, Android OS, this work describes an analytical SAR model in the form of continuous-time Markov chains (CTMC) [11].

The object of the study is the process of software aging. The subject of the study is a SAR model for mobile OS. The purpose of the work is to develop a SAR model for Android OS considering mobile device usage by users and different aging levels.

Related works

Software aging phenomenon is observed in mobile systems, in particular, Android OS [7-9]. Empirical studies results allow to determine system components that are vulnerable to the software aging [8, 12, 13] and to identify effective aging-related indicators [8, 12, 14]. Developed analytical models allow to predict the aging process and determine the optimal time to perform the rejuvenation procedure [15].

An aging detection and rejuvenation agent for Android (ADARTA) [16] measures aging-related metrics and, analysing the obtained time series, detects software aging, plans and performs software rejuvenation. ADARTA implements a moving window method for tracking and detecting software aging in real time and requires constant use of system resources to perform calculations. Therefore, the advantage of using analytical models to predict aging and plan the rejuvenation procedure is that they may decrease calculations in the background system process.

This paper [17] describes a general SAR model based on Markov Chains considering additional states to describe the gradual deterioration of system performance: “Robust”, “High Efficiency”, “Medium Efficiency”, “Low Efficiency”, “Unstable”. This model allows to perform different types of rejuvenation procedures in the "Unstable" state, which restore the system to one of the previous states in accordance with the effectiveness of a particular type of rejuvenation procedure. The implementation of this model allows to determine the optimal rejuvenation strategy and mechanism that will ensure the greatest system availability.

System performance levels in models allow to take into account the advantages of measurements-based and thresholds-based approaches [5, 18]. For example, the following aging indicators [13, 14] were experimentally tested in previous work for Android OS, average or median values of which can be used to determine the aging status of the system at a certain point in time:

- Frame Draw Time (FDT) is a time for displaying user interface (UI) frame of the application, or otherwise this metric can be interpreted as FPS;
- Proportional Set Size (PSS) is the amount of RAM used by individual processes in the system.

Taking into account of the rejuvenation procedure mechanism [19, 20] in the SAR model is an important task, because the concrete implementation of this procedure may have a negative impact on the user experience. The device and its functions may be partially or completely inaccessible to the user during the rejuvenation procedure.

There are two main groups of rejuvenation mechanisms [20]: cold and warm rejuvenation. Cold rejuvenation is a complete reboot of a mobile device, which means that the system is unavailable during this procedure, but reboot provides the best recovery of system performance. In turn, warm rejuvenation consists in rebooting or cleaning system services, user applications and processes, so it can be assumed that the system is still available for user. However, warm rejuvenation is less effective than cold rejuvenation, and its effectiveness depends on its specific implementation and aging conditions.

Analytical model of SAR for Android OS [15] based on CTMC takes into account mobile device usage as two additional states of the system, namely “Active” and “Sleep” states. This model can be extended and improved because it does not take into account different aging levels and types of rejuvenation procedures.

Materials and methods

This paper proposes an extended SAR model for the Android OS based on CTMC, which takes into account the usage activity of a mobile device, the additional aging level of the system, and provides either a cold or warm rejuvenation mechanism.

First of all, it is necessary to describe a general extended SAR model considering aging levels in the form of CTMC, which consists of five states: “Young”, “Aging”, “Old”, “Rejuvenation”, “Failure”, where “Young”, “Aging”, and “Old” states reflect a gradual degradation of the system performance in terms of aging. Thus, the extended SAR model can be presented in the form of states and transitions graph, as shown in fig. 1, where the states are described as follows:

1) “Young” (Y) is a system state, which is characterized by a high level of productivity and low level of resource consumption, for example, the average or median values of FDT and PSS system processes are not higher than the set thresholds;

2) “Aging” (A) is a system state in which there is a performance deterioration and increased consumption of system resources, but it has little effect on user experience (UX), in particular, the average or median values of FDT and PSS of system processes are within predetermined limits;

3) “Old” (O) is a system state when the user experiences delay in the UI, and exhaustion of system resources can lead to failures in user applications: the average or median values of FDT and PSS system processes are higher than the set thresholds;

4) “Rejuvenation” (R) is a system state in which the rejuvenation procedure takes place. The system may be unavailable in this state if cold rejuvenation is performing;

5) “Failure” (F) is an aging-related failure state in which the system recovers. Recovery consists in mobile device rebooting by the system itself or user.

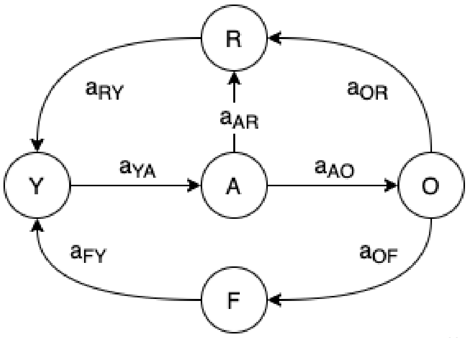


Fig.1. CTMC for the extended general SAR model considering different aging levels

To calculate the transition rates between aging states (a_{ij}) as well as transition times (T_{ij}), it is important to consider the presence or absence of constant trends of metric deterioration in these states, namely in “Young”, “Aging”, and “Old”. For example, a system in the “Young” or “Aging” state may not show increasing trends of FDT, so the transition to the “Aging” and “Old” state may not occur. In this case the rejuvenation procedure, as well as the need to predict its optimal time may be impractical.

Thus, in contrast to the original model [15], the additional “Aging” state in the proposed model allows to clearly distinguish the aging process in the system and distinguish it from actually “Old” state. In this case, the extended model provides the ability to perform rejuvenation in both “Aging” and “Old” states, which is described by the a_{AR} and a_{OR} transition rates, respectively.

The extended SAR model for mobile OS is presented in fig. 2 in the form of a states and transitions graph of the Markov Chain. The model takes into account both the mobile device usage activity and additional software aging levels.

States of proposed SAR model are follows: AmY is an “Active mobile Young” state; AmA is an “Active mobile Aging” state; AmO is an “Active mobile Old” state; AmR is an “Active mobile Rejuvenation” state; AmF is an “Active mobile Failure” state; SmY is a “Sleep mobile Young” state; SmA is a “Sleep mobile Aging” state; SmO is a “Sleep mobile Old” state; SmR is a “Sleep mobile Rejuvenation” state; SmF is a “Sleep mobile Failure” state.

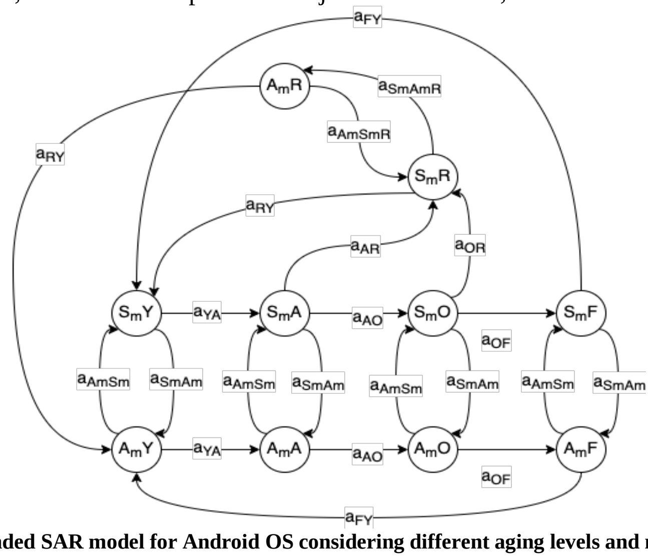


Fig.2. CTMC for the extended SAR model for Android OS considering different aging levels and mobile device usage activity

Software rejuvenation planning is about choosing the optimal time to perform it. The optimal time from the user's point of view is the state of the system when the mobile device is not used, namely, in the states SmY, SmA, SmO. In turn, the optimal time from the system's point of view is the time when there is a productivity deterioration, rejuvenation procedure performs before aging-related failure, as well as rejuvenation does not interrupt performing of priority processes, i.e., in SmA and SmO states. Therefore, the optimal rejuvenation time for the user and the system is possible in the SmA and SmO states.

The proposed model allows to consider three main strategies for rejuvenation planning, namely:

1) Simultaneous rejuvenation from "Aging" and "Old" states, i.e., when $a_{AR} > 0.0$ and $a_{OR} > 0.0$;
2) Planning rejuvenation only from the "Aging" state, when $a_{AR} > 0.0$ and $a_{OR} = 0.0$;
3) Planning rejuvenation only from the "Old" state, when $a_{AR} = 0.0$ and $a_{OR} > 0.0$.

Given the existence of different rejuvenation mechanisms, namely cold and warm rejuvenation, the model introduced two additional transition rates between the AmR and SmR states. These parameters allow to determine which rejuvenation method is used calculating the model. The condition, when $a_{SmAmR} = 0$ and $a_{AmSmR} = 0$, means that the user will not be able to use the mobile device while cold rejuvenation procedure performing. Thus, taking into account the mobile device usage factor, an important task is to choose the optimal cold or warm rejuvenation for a specific scenario of mobile device use.

Since the SAR process is modelled as a CTMC, the probability of the system being in the *i*-th state $P_i(t)$ is obtained from the solution of the system of Kolmogorov – Chapman equations:

$$\frac{dP_i(t)}{dt} = -\sum_{j\in S} a_{ij}P_i(t) + \sum_{j\in S} a_{ji}P_j(t)\,, i \in S, \tag{1}$$

where $P_i(t)$ ($P_j(t)$) – is the probability of a system being at time t in the i(j) state, a_{ij} (a_{ji}) – is a transition rate from i state to j state (from j state to i state) at time t, S – is a set of all system states.

A Kolmogorov-Chapman system of differential equations for proposed SAR model:

$$\begin{aligned}
&\frac{dP_{SmY}(t)}{dt} = -(a_{YA} + a_{SmAm})P_{SmY}(t) + a_{FY}P_{SmF}(t) + a_{AmSm}P_{AmY}(t) + a_{RY}P_{SmR}(t);\\
&\frac{dP_{SmA}(t)}{dt} = -(a_{AO} + a_{SmAm} + a_{AR})P_{SmA}(t) + a_{YA}P_{SmY}(t) + a_{AmSm}P_{AmA}(t);\\
&\frac{dP_{SmO}(t)}{dt} = -(a_{OF} + a_{OR} + a_{SmAm})P_{SmO}(t) + a_{AO}P_{SmA}(t) + a_{AmSm}P_{AmO}(t);\\
&\frac{dP_{SmF}(t)}{dt} = -(a_{FY} + a_{SmAm})P_{SmF}(t) + a_{OF}P_{SmO}(t) + a_{AmSm}P_{AmF}(t);\\
&\frac{dP_{AmY}(t)}{dt} = -(a_{YA} + a_{AmSm})P_{AmY}(t) + a_{FY}P_{AmF}(t) + a_{RY}P_{AmR}(t) + a_{SmAm}P_{SmY}(t);\\
&\frac{dP_{AmA}(t)}{dt} = -(a_{AO} + a_{AmSm})P_{AmA}(t) + a_{YA}P_{AmY}(t) + a_{SmAm}P_{SmA}(t);\\
&\frac{dP_{AmO}(t)}{dt} = -(a_{OF} + a_{AmSm})P_{AmO}(t) + a_{AO}P_{AmA}(t) + a_{SmAm}P_{SmO}(t);\\
&\frac{dP_{AmF}(t)}{dt} = -(a_{FY} + a_{AmSm})P_{AmF}(t) + a_{OF}P_{AmO}(t) + a_{SmAm}P_{SmF}(t);\\
&dPSmR(t)dt=-(aRY+aSmAmR)PSmR(t)+aARPSmA(t)+aORPSmO(t)+aAmSmRPAmR(t);\\
&\frac{dP_{AmR}(t)}{dt} = -(a_{RY} + a_{AmSmR})P_{AmR}(t) + a_{SmAmR}P_{SmR}(t).
\end{aligned} \tag{2}$$

A system of differential equations (2) can be calculated by numerical methods, in particular by the Runge-Kutta method.

Since the time distribution in models based on the CTMC is exponential, the average transition time to a certain state is determined by the formula:

$$T_{avg} = \frac{1}{a}, \tag{3}$$

where a – is a transition rate between states.

Experiments

To verify and analyse the proposed extended SAR model, the system of differential equations (2) was calculated by the 4th order Runge-Kutta method. Average transition times between all states of the system are approximate values which allow to simulate aging process and perform experimental calculations. Table 1 presents a set of test simulations for different conditions of software aging, intensities of mobile device usage, and types of rejuvenation procedure.

The mobile device usage model is represented by two sets of transitions between states that describe different levels of user activity during the day:

- Moderate use, where T_{SmAm} = 75 minutes (a_{SmAm} = 0.013), and T_{AmSm} = 10 minutes (a_{AmSm} = 0.1);
- Active use, where T_{SmAm} = 30 minutes (a_{SmAm} = 0.033), and T_{AmSm} = 50 minutes (a_{AmSm} = 0.02).

The software aging model describes two devices that have different performance and aging rates, namely:

- A device with high performance and resistance to software aging, where T_{YA} = 48 hours (a_{YA} = 0.00035), T_{AO} = 96 hours (a_{AO} = 0.00018), T_{OF} = 24 hours (a_{OF} = 0.00069), and T_{FY} = 1 minute (a_{FY} = 1.0);

- A device with low productivity and vulnerability to software aging, where T_{YA} = 4 hours (a_{YA} = 0.0042), T_{AO} = 8 hours (a_{AO} = 0.0021), T_{OF} = 2 hours (a_{OF} = 0.0083), and T_{FY} = 5 minutes (a_{FY} = 0.2).

Table 1

Set of test simulations for different conditions

SIM	Model of mobile device usage		Model of software aging				Model of software rejuvenation				
	T_{SmAm}, min	T_{AmSm}, min	T_{YA}, hours	T_{AO}, hours	T_{OF}, hours	T_{FY}, min	T_{AR}, hours	T_{OR}, hours	T_{RY}, min	T_{SmAmR}, min	T_{SmAmR}, min
SIM-0	75	10	48	96	24	-	-	-	-	-	-
SIM-1	75	10	48	96	24	1	-	-	-	T_{SmAm}	T_{AmSm}
SIM-2	75	10	4	8	2	5	-	-	-	T_{SmAm}	T_{AmSm}
SIM-3	30	50	48	96	24	1	-	-	-	T_{SmAm}	T_{AmSm}
SIM-4	30	50	4	8	2	5	-	-	-	T_{SmAm}	T_{AmSm}
SIM-5	75	10	48	96	24	1	48	12	1	T_{SmAm}	T_{AmSm}
SIM-6	75	10	48	96	24	1	48	-	1	T_{SmAm}	T_{AmSm}
SIM-7	75	10	48	96	24	1	-	12	1	T_{SmAm}	T_{AmSm}
SIM-8	75	10	48	96	24	1	144	36	1	T_{SmAm}	T_{AmSm}
SIM-9	75	10	4	8	2	5	4	1	5	T_{SmAm}	T_{AmSm}
SIM-10	75	10	4	8	2	5	4	-	5	T_{SmAm}	T_{AmSm}
SIM-11	75	10	4	8	2	5	-	1	5	T_{SmAm}	T_{AmSm}
SIM-12	75	10	4	8	2	5	12	3	5	T_{SmAm}	T_{AmSm}
SIM-13	30	50	48	96	24	1	48	12	1	T_{SmAm}	T_{AmSm}
SIM-14	30	50	48	96	24	1	48	-	1	T_{SmAm}	T_{AmSm}
SIM-15	30	50	48	96	24	1	-	12	1	T_{SmAm}	T_{AmSm}
SIM-16	30	50	48	96	24	1	144	36	1	T_{SmAm}	T_{AmSm}
SIM-17	30	50	4	8	2	5	4	1	5	T_{SmAm}	T_{AmSm}
SIM-18	30	50	4	8	2	5	4	-	5	T_{SmAm}	T_{AmSm}
SIM-19	30	50	4	8	2	5	-	1	5	T_{SmAm}	T_{AmSm}
SIM-20	30	50	4	8	2	5	12	3	5	T_{SmAm}	T_{AmSm}
SIM-21	75	10	48	96	24	1	48	-	1	-	-
SIM-22	75	10	4	8	2	5	4	-	5	-	-
SIM-23	30	50	48	96	24	1	48	-	1	-	-
SIM-24	30	50	4	8	2	5	4	-	5	-	-

This paper considers two options of aging-related failures, the comparison of which allows assessing the impact of recovery after the aging-related failure on the behaviour of the system:

- Aging-related failure causes the mobile device reboot which returns the system to the "Young" state, i.e., the system is in the FY state only the time required to reboot;
- Aging-related failure does not return the system to the "Young" state, i.e., T_{FY} = 0 minutes (a_{FY} = 0.0).

Rejuvenation strategies are represented by two data sets for two devices with different software aging rates, respectively.

The rejuvenation procedure for a device with high performance and resistance to software aging lasts on average 1 minute (a_{RY} = 1.0), and planning strategies are presented by the following sets of T_{avg} and a_{ij}:

- Rejuvenation in "Aging" and "Old" states, where T_{AR} = 48 hours (a_{AR} = 0.00035), T_{OR} = 12 hours (a_{OR} = 0.0014);
- Rejuvenation in "Aging" state, where T_{AR} = 48 hours (a_{AR} = 0.00035), T_{OR} = 0 hours (a_{OR} = 0.0);
- Rejuvenation in "Old" state, where T_{AR} = 0 hours (a_{AR} = 0.0), T_{OR} = 12 hours (a_{OR} = 0.0014);
- Rejuvenation is performed with a delay in "Aging" and "Old" states, where T_{AR} = 144 hours (a_{AR} = 0.00012), T_{OR} = 36 hours (a_{OR} = 0.00046).

The rejuvenation procedure for a device with low performance and vulnerability to aging lasts an average of 5 minutes (a_{RY} = 0.2), and planning strategies are presented by the following sets of T_{avg} and a_{ij}:

- Rejuvenation in "Aging" and "Old" states, where T_{AR} = 4 hours (a_{AR} = 0.0042), T_{OR} = 1 hour (a_{OR} = 0.017);
- Rejuvenation in "Aging" state, where T_{AR} = 4 hours (a_{AR} = 0.0042), T_{OR} = 0 hours (a_{OR} = 0.0);
- Rejuvenation in "Old" state, where T_{AR} = 0 hours (a_{AR} = 0.0), T_{OR} = 1 hour (a_{OR} = 0.017);
- Rejuvenation is performed with a delay in "Aging" and "Old" states, where T_{AR} = 12 hours (a_{AR} = 0.0014), T_{OR} = 3 hours (a_{OR} = 0.0056).

To assess the impact of recovery after aging-related failure, the model was calculated for SIM-0, which is identical to SIM-1, but has no transition from "Failure" state to "Young" state (a_{FY} = 0.0), i.e., aging-related failure is a terminal state of the system and recovery does not occur immediately.

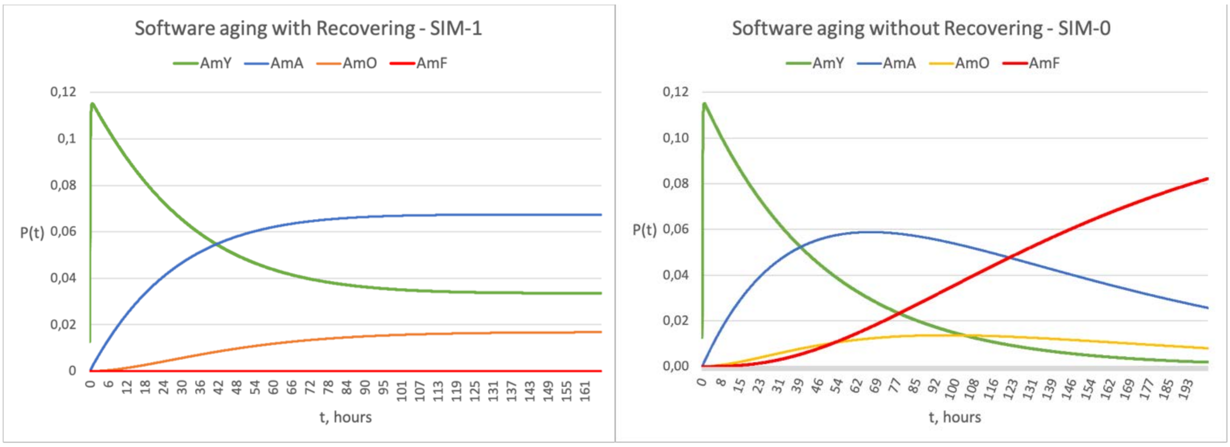
To test the extended SAR model and evaluate the aging process, the model was calculated without rejuvenation in SIM-0 - SIM-4 simulations. The impact of software aging process on UX can be determined using estimates $P_{AmY}(t)$, $P_{AmA}(t)$, $P_{AmO}(t)$ і $P_{AmF}(t)$, i.e., the probabilities of the system at different aging levels during active usage of the mobile device by the user.

To analyze the rejuvenation planning strategies, simulation sets were calculated for different combinations of mobile devices and the activity of their use (SIM-5 - SIM-20). The obtained results of simulations are also compared with simulations without rejuvenation (SIM-1 - SIM-4). The effectiveness of the rejuvenation procedure in different strategies can be assessed by the probability of system beeing in the AmY state, i.e., the most effective strategy provides the highest probability of active use of the mobile device in the high-performance state.

To analyze the mechanisms of a cold and warm rejuvenation, additional simulations SIM-21 - SIM-24 were performed, where cold rejuvenation is performed, i.e., $T_{SmAmR} = T_{AmSmR} = 0.0$. In turn, simulations SIM-1 - SIM-20 describe warm rejuvenation, where $T_{SmAmR} = T_{SmAm}$, $T_{AmSmR} = T_{AmSm}$.

Results

Fig. 3 shows calculation of SIM-0 and SIM-1 when an inactive user is using a mobile device that is aging slowly. A comparison of these simulations allows to characterize the impact of recovering after aging, as well as emphasize the importance of applying the rejuvenation procedure to improve UX and increase the uptime of the mobile device.



b)

Fig.3. Comparison of software aging models without rejuvenation: a) with recovery after aging-related failure; b) without recovery after aging-related failure.

In fig. 3a, the probabilities of all states reach the steady level after the 100th hour of the mobile device use. This behaviour of the model is due to the presence of the “Failure” state, in which there is a recovery with a return to the “Young” state. Comparison of SIM-0 and SIM-1 allows to conclude that without any reaction to the aging-related failure (fig. 4b), the probability of being in a young state AmY goes to 0, and the aging process (i.e., changes of AmY, AmA, AmO, and AmF) leads to complete system failure. In the case of SIM-1 (fig. 3a), the mobile device will necessarily go to the so-called “recovery plateau”, which provides a return to a more productive system state, but forces the user or system to reboot the mobile device.

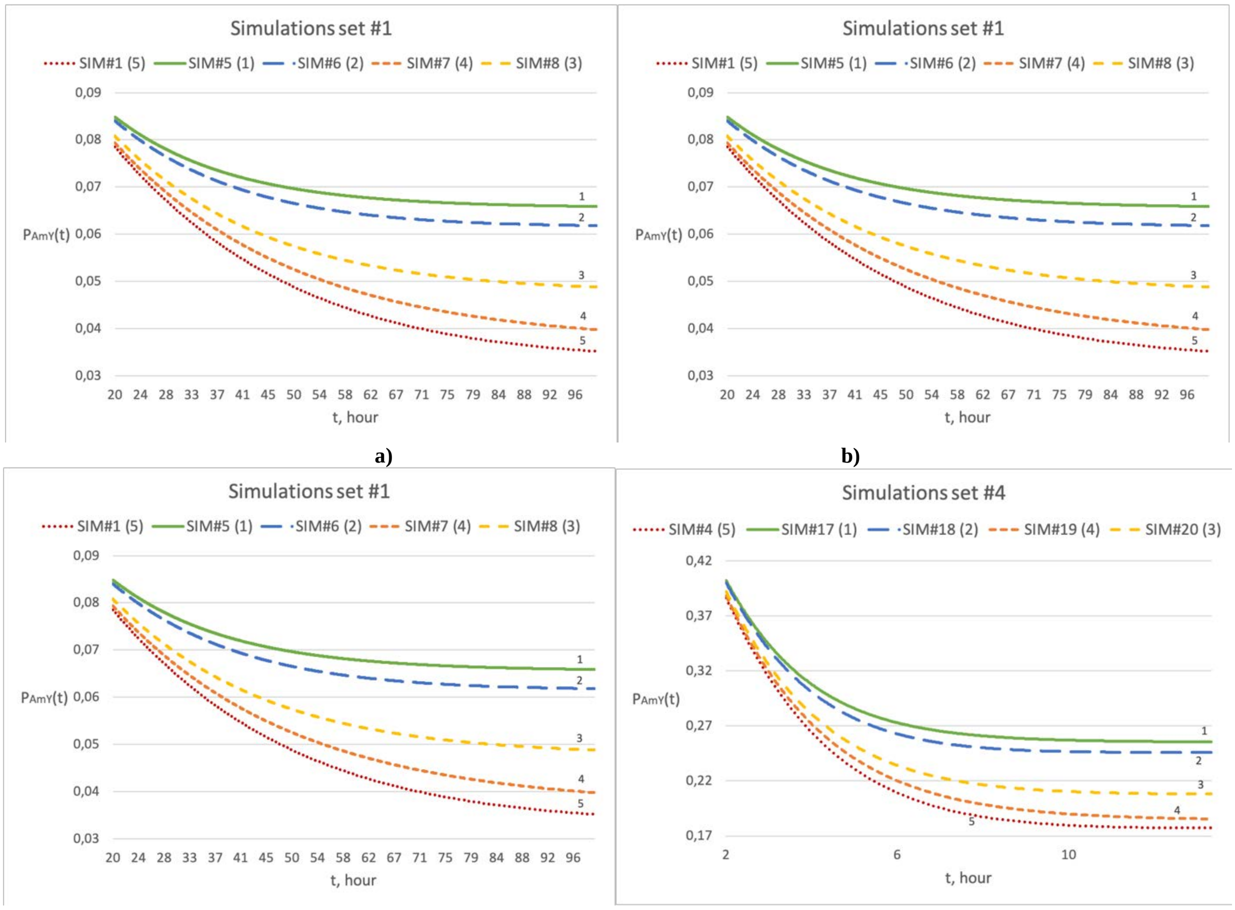
Thus, the performed software aging simulations show that the important tasks of the rejuvenation procedure to improve UX are:

- increase the mobile device availability: $max(P_{AmY}(t))$;
- delay aging-related failures: $min(P_{AmA}(t) + P_{AmO}(t) + P_{AmF}(t))$.

Fig. 4 shows an effectiveness comparison of different strategies for four sets of simulations. Each set describes different rejuvenation strategies for the same aging conditions. The curves on the graphs are numbered from 1 to 5 in descending order of efficiency.

Simultaneous rejuvenation from the “Aging” and “Old” states allows to ensure the highest level of the mobile device availability in the “Young” state (curves (1) in fig. 4). A less effective strategy is to plan rejuvenation only from the “Aging” state (curves (2) in fig. 4). The worst strategy is to plan rejuvenation only from the “Old” state (curves (4) in fig. 4). The absence of rejuvenation is expected to show the lowest values (curves (5) in fig. 4). Also, it can be seen that simulations in which rejuvenation is scheduled with a delay for both the “Aging” and “Old” states (curves (3) in fig. 4) show better efficiency than simulations without rejuvenation or with rejuvenation only in the “Old” state.

The obtained results confirm the effectiveness of premature rejuvenation in the “Aging” state, before the transition to the “Old” state. In this case, to ensure high availability of the mobile device, it is necessary to repeat the rejuvenation procedure. Although delayed rejuvenation (curves (3) in fig. 4) increases the likelihood of being in the “Young” state, however, this case is not optimal, so it is important to choose the optimal time between the transition from “Aging” to “Old”.



a) b) c) d)

Fig.4. An effectiveness comparison of different rejuvenation strategies for four sets of simulations: a) simulations set №1; b) simulations set №2; c) simulations set №3; d) simulations set №4.

Comparison of SIM-24 and SIM-18 simulations with the same aging conditions, but different rejuvenation mechanics is shown in fig. 5. Curve number 3 shows the percentage increase in the probability of system being in the "Young" state if warm rejuvenation is performed.

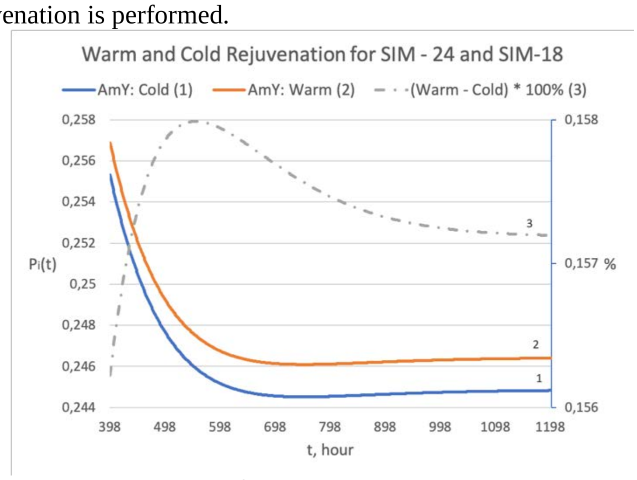


Fig.5. Comparison of warm and cold rejuvenation methods

Fig. 5 shows that in case of warm rejuvenation, the device is indeed more likely to be in a AmY state than in the case of cold rejuvenation. SIM-24 and SIM-18 simulations describe the case when an active user uses a mobile device that is aging rapidly, so in this case, the important tasks of rejuvenation are to ensure both the mobile device availability and the effectiveness of chosen strategy. Warm rejuvenation can provide availability, however, if the device performs warm rejuvenation in the "Old" state, then the probability of restoring the system to the "Young" state may be extremely small due to the peculiarities of this approach. Thus, the proposed SAR model with warm rejuvenation cannot be applied to the case of performing rejuvenation in the "Old" state without the assumption that warm procedure will restore the system to the "Young" state. In other words, it can be assumed that warm rejuvenation method in the proposed model can be effective only in the case of planning rejuvenation in the "Aging" state, and cold rejuvenation method can be effective in both "Aging" and "Old" states.

Conclusions

Software aging has a negative impact on UX and software reliability. An effective method to counteract software aging is to schedule and perform the software rejuvenation procedure. An important task is to develop SAR models considering features of mobile devices which can be employed to select optimal strategy and time of rejuvenation.

The scientific novelty of the proposed mathematical SAR model for mobile device is to consider additional software aging levels and warm or cold rejuvenation mechanisms. The developed model is represented by a CTMC, which combines the mobile device usage model and the extended SAR model.

The practical significance of using additional software aging levels in the model consists in the possibility of applying methods based on measurements and thresholds of aging-related metrics. The results of the simulations showed that an effective strategy for software rejuvenation is its repeated execution in the "Aging" state. Proposed SAR model allows to perform rejuvenation, when necessary, i.e., when the software aging is observed in the system, after "Young" state. At the same time, the model allows to plan rejuvenation in advance of the transition to a low-productivity state, in which there is higher probability of aging-related failure, i.e., before "Old" state. When planning a rejuvenation procedure, considered properties of proposed model may allow to improve UX as well as device reliability. Using model, it is possible to select optimal rejuvenation mechanism (namely, cold or warm rejuvenation) for specific conditions of device using, which has practical value for improving UX.

The future research is to investigate proposed model for different conditions and real data, because in this work a simulation was performed using test approximate values of transition times. It is important to investigate cold and warm rejuvenation in different usage scenarios, in particular, the use of cold and warm rejuvenation from different aging levels and with different effects on improving system performance. Also, proposed SAR model can be employed to implement rejuvenation method for Android OS and considered rejuvenation strategies should be verified in developed methods.

Acknowledgements

The work is supported by the Ukrainian national state budget research project number 0121U109527.

References

1. ISO 25010. – https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&limitstart=0/ [Accessed 17 November 2021].
2. Parnas D. L. Software aging. *16th International Conference on Software Engineering* : proceedings, Sorrento, Italy, 1994. P.279-287. DOI: https://doi.org/10.1109/ICSE.1994.296790
3. Huang Y., Kintala C., Kolettis N., Fulton N. D. Software rejuvenation: Analysis, module and applications, *25th Symposium on Fault Tolerant Computing* : proceedings, Pasadena, 27-30 June 1995. Pasadena, California, 1995. P.381–390. DOI: https://doi.org/10.1109/FTCS.1995.466961
4. Grottke M., Jr R. M., Trivedi K. S. The fundamentals of software aging. *IEEE 19th International Symposium on Software Reliability Engineering* : proceedings. – 2008. – P.1–6.
5. Cotroneo D., Natella R., Pietrantuono R., Russo S. A Survey of Software Aging and Rejuvenation Studies *ACM Journal on Emerging Technologies in Computing Systems*. 2014. Vol. 10. No 1. Article 8. DOI: https://doi.org/10.1145/2539117
6. Valentim N. A., Macedo A., Matias R. A Systematic Mapping Review of the First 20 Years of Software Aging and Rejuvenation Research. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2016. DOI: https://doi.org/10.1109/ISSREW.2016.42
7. Maji A. K., Hao K., Sultana S., Bagchi S. Characterizing failures in mobile OSes: a case study with Android and Symbian. *International Symposium on Software Reliability Engineering. IEEE Computer Society*. 2010. P. 249-258. DOI: https://doi.org/10.1109/ISSRE.2010.45
8. Cotroneo D., Fucci F., Iannillo A. K., Natella R., Pietrantuono R. Software aging analysis of the android mobile os. *IEEE 27th International Symposium on Software Reliability Engineering*. 2016. P. 478-489. DOI: https://doi.org/10.1109/ISSRE.2016.25
9. Yakovyna V.S., Uhrynovskyi B. V. Software aging in the context of reliability: a review of the issue. *Scientific Bulletin of UNFU.* 2020. №30 (2). P. 107-112. DOI: https://doi.org/10.36930/40300219
10. Ahamad S. Study of software aging issues and prevention solutions. *The International Journal of Computer Science and Information Security*. 2016. Vol. 14, №8. P. 307-313.
11. Norris J. R. Markov Chains. *Cambridge University Press*, 1997.
12. Cotroneo D., Iannillo A. K., Natella R., Pietrantuono R. A Comprehensive Study on Software Aging across Android Versions and Vendors. *Empirical Software Engineering*. 2020. Vol. 25. P. 3357-3395. DOI: https://doi.org/10.1007/s10664-020-09838-3
13. Yakovyna V.S., Uhrynovskyi B.V. Research of software aging in Android system processes and user applications. *Herald of Khmelnytskyi national university*. 2021. №2 (295). P. 64-70. DOI: https://www.doi.org/10.31891/2307-5732-2021-295-2-64-70
14. Yakovyna V.S., Uhrynovskyi B.V. User-perceived Response Metrics in Android Os for Software Aging Detection. *Journal of Lviv Polytechnic National University "Information Systems and Networks"*. 2021. Vol. 9. P. 32-43. DOI: https://doi.org/10.23939/sisn2021.09.032
15. Xianga J., Wenga C., Zhaoa D., Tiana J., Xionga S., Lia L., Andrzejak A. A New Software Rejuvenation Model for Android. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2018. DOI: https://doi.org/10.1109/ISSREW.2018.00021
16. Cotroneo D., Simone L.D., Natella R., Pietrantuono R., Russo S. A Configurable Software Aging Detection and Rejuvenation Agent for Android. *11th Intl Workshop on Software Aging and Rejuvenation (WoSAR)*. 2019. DOI: https://doi.org/10.1109/ISSREW.2019.00078
17. Ghobadi Z. R., Rashidi H. A software availability model based on multilevel software rejuvenation and markov chain. *Turkish journal of Electrical Engineering & Computer Sciences*. 2021. Vol. 29(2). P. 730-744. DOI: https://doi.org/10.3906/elk-2003-159
18. Yakovyna V.S., Uhrynovskyi B.V. Software aging in the context of its reliability: a systematic review. *Scientific Bulletin of UNFU.* 2019. №29 (5). P. 123-128. DOI: https://doi.org/10.15421/40290525
19. Weng C., Zhao D., Lu L., Xiang J., Yang C., Li D. A Rejuvenation Strategy in Android. *IEEE International Symposium on Software*

Reliability Engineering Workshops (ISSREW). 2017. P. 273-279. DOI: https://doi.org/10.1109/ISSREW.2017.50
20. Guo C., Wu H., Hua X., Lautner D., Ren S. Use Two-Level Rejuvenation to Combat Software Aging and Maximize Average Resource Performance. *IEEE 12th International Conference on Embedded Software and Systems*. 2015. P. 1160-1165.

Vitaliy Yakovyna Віталій Яковина	DrS, Full Professor of Artificial Intelligence Department, Lviv Polytechnic National University, Lviv, Ukraine. e-mail: vitaliy.s.yakovyna@lpnu.ua. orcid.org/0000-0003-0133-8591, Scopus Author ID: 6602569305, ResearcherID: N-8402-2017, https://scholar.google.com.ua/citations?hl=uk&user=e1Sj2JYAAAAJ	доктор техн. наук, проф. кафедри систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.
Bohdan Uhrynovskyi Богдан Угриновський	Assistant of Software Department, Lviv Polytechnic National University, Lviv, Ukraine. e-mail: bohdan.v.uhrynovskyi@lpnu.ua. orcid.org/0000-0002-4356-192X, Scopus Author ID: 57215328321, https://scholar.google.com/citations?user=dkI33_kAAAAJ&hl=uk&authuser=1&oi=ao	асистент кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.