Service Quality Meter Design for Self-Aware Systems

Kalle Tammemäe[1] and Mairo Leier[2]

Abstract— **Different resource meters are part of almost all human invented devices and gadgets. Typical known examples are battery level indicators of mobile devices or gasoline level indicators of cars. In more complex systems there might be tens or hundreds of indicators telling status of different critical resources, remaining expected lifetime of critical components or just readiness of service (heartbeat) at the instant. Logging of set of important parameters is often used for off-line analysis the service quality of the systems, in some of cases such post-analysis is extremely important to discover exact reason of catastrophic failure like in case of black boxes of planes and other large vehicles. Still, as it is known, the observable history of black boxes is limited, covering only limited number of last service hours. Contemporary systems have started to obtain certain level of (self-) awareness, meaning ability to assess development of internal and external world according to certain fixed of flexible self-trainable model. Such systems are able to predict own ability and quality to fulfill defined goals in the near future on base of past service history and current situation. The paper is analyzing possible indicator designs to represent history of service over system full lifetime, averaging far past events to limit information stored in non-volatile memory. To reach that different classification scenarios are used to discriminate regular events and anomalous (critical) ones which are requiring special attention because might negatively impact probability to fulfill defined goals in future. The running pattern of system past life could be a part of self-awareness property of the system itself, i.e. additional feedback to increase precision of prediction the service quality in the near future.**

I. INTRODUCTION

Contemporary systems are supplied with various meters to supply users with information about specific vital resource level or in case of some additional processing with expected endurance of specific resource. E.g. on an electronic dashboard of car driver can see how many km is possible to cover with current driving profile, mobile device is warning user when battery level is below specific value (e.g. 10% of charge) and can switch device to sleep mode before remaining battery level is becoming too critical to save work status. Engines have hard-wired hour-meters counting working hours and warning users about approaching the service interval term. Computer gamers follow carefully own life meters, like presented in Figure 1, running in maze to find "food", "medicine packages" and to avoid "life-shortening" damage to their avatar.

It is also common to record system lifetime (working hours/days) using hour-meters to inform user about need to visit service for regular engine/car maintenance. Planes and other large vehicles are equipped with black boxes to record number of system parameters for later off-line analysis, like in Figure 2. Such kind of information is especially useful to discover reason of malfunction or catastrophic event. Recording everything is itself quite a resource consuming, and due to that time-limited. E.g. in case of lost Malaysian airplane flight MH370 - it is known that once the recorders will be found, they do not contain discussions in cockpit at time the plane started to behave mysteriously - for that the cockpit voice recording quota is too small, the recording loop is covering only 2 last hours. Still, the flight data is recorded in black box over full flight[3]. Black boxes of car are default equipment in almost all newer models (http://www.carblackbox.co.uk/). One of possible solutions to record data before and whenever possible during and after crash situation is provided in paper [1]. The proposed design is intended to keep 200 seconds data record in real-time. Vehicle tachographs can record data, e.g. speed and driving mode, over longer period but do not contain information about vehicle actual "health".

Event logging of complex system during their full lifetime is producing enormous amount of data, analysis of which can be done only off-line using contemporary databases and data analysis tools. In consumer market such appliance lifetime behavior information is not collected. E.g. looking for used car buyers only source of information is service book (or records in car service), there is no numerical information how well or badly car and its engine behaved in hands of former owner(s). In some cases to keep recorded data over long period of time, some of the data is averaged over time. The older the data is, the more it will be averaged until

Fig. 1. Life-meter of computer game. Hearts are often associated with health or extra lives (http://www.giantbomb.com/heart/3055-142/images/)

[1]Department of CE, Tallinn University of Technology, Tallinn, Estonia kalle.tammemae@ttu.ee

[2]Department of CE, Tallinn University of Technology, Tallinn, Estonia mairo.leier@ttu.ee

[3]http://www.telegraph.co.uk/technology/news/10726184/Malaysia-Airlines-MH370-the-black-box-explained.html

Fig. 2. Cockpit Voice and Flight Data Recorder. Source: http://www.l-3ar.com/

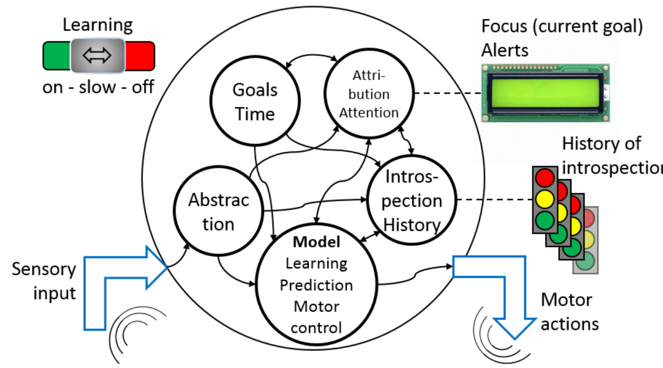

Fig. 3. General self-aware system architecture

the data over once per second becomes to over minute, day or even month. Even when most of the data is lost it is still possible to use averaged historical data for rough estimation.

Continuous advance in computing technology and software is increasing embedded intelligence of contemporary systems. Different awareness aspects are discussed in research papers like power-awareness, situation awareness, self-awareness among others [2]. Self-awareness itself is concerned with the availability, collection and representation of knowledge about a system, by that system [3]. Some computational resource of the system could be used to make robust real-time assessment and even prediction of service quality of the system in near future. In principle, decreasing price of energy-independent memory allows recording of behavioral data of the system during very long service periods. Still, the last property remains probably limited and is also not the way biological (living) systems behave. Like in case of us, humans, only anomalous (emotionally remarkable) situations are remembered firmly together with preceding and following situation details to avoid or *vice versa* to recall the same situation in the future. Concerning exact time then it is memorized rather relatively to other daily or seasonal events - we remember what another large event happened before or after specific event but barely are able to fix it in calendar precisely. Moreover, as everyone has experienced - really old events are getting fuzzy and deteriorate, their accuracy is fading and we are able recall in details only the brightest memories. Despite that feature all living beings are able to make reasonable decisions based on learned experience for their best survival.

In [4, figure 2, page 41], the architecture of a prototype self-aware health monitor was provided. In details such hypothetical system on Figure 3 consists of number of internal and external components of which we concentrate on representation of introspection history. In self-aware systems with many feedback loops crucial role is played by attention, which controls the whole system activity - proceed normally or in case of irregularities in sensory data (anomaly) study the situation, correct model and alert user(s). Such system is continuously observing external environment and its own behavior i.e. ability or inability to fulfill predefined tasks, resulting a real-time self-assessment. If anomalies are small and do not cause deviation of current goal pursuing path, then the self-assessment grade is positive. In case of serious anomalies (e.g. due to out of regular range sensory input) causing interruption of current tasks and seeking for external help from user(s) the self-assessment grade will be rather negative.

II. Hypothesis

To get overview over system service lifetime, partial information is sufficient whereas more details (rather increasingly) are expected to be available about recent behavior. Moreover, instead of behavior indicators (current focus on Figure 3) the assessment of behaviour in respect of goal achievement is more informative. The idea in this paper is to redesign set of classical resource level meters to reflect introspection as service quality in respect of predefined temporal goals over system lifetime. When the system is behaving (conditions are met) according to set of goals then assessment is positive. In case some signals (external and/or internal) are disturbing goal-matching behaviour then assessment is negative. The assessment is task of introspection unit which is examining the current state and behavior of the system in respect of current goal and goal defined tasks. Instead of presenting to user only current assessment we propose to preserve digest of past behaviour over long (full) lifetime of the system.

III. First (instant) and second (full or fading history) level of self-assessment

We propose for representing of self-assessment to use some one-dimensional array of graphic display elements or individual LED-indicators. In real-life the number of visualization elements is limited. To represent self-assessment over time, lets denote sequence of indicators with n elements vector $|D| = d_{n-1}, d_{n-2}, \ldots, d_0$, where every d_i is representing certain average over past self-assessment values and d_0 is element to represent the recent self-assessment value. Lets denote current self-assessment value: g^t. We propose lifeline to be arranged so that the rightmost element of lifeline is visualizing current self-assessment value g^t and rest of elements average over values in history $g^{t-1}, g^{t-2}, \ldots$

Fig. 4. Example color schema to represent system self-assessment value

in some color coded scheme as for example in Figure 4. There has to be mapping from $G \to C$, wher C is discrete set of colors.

We expect some logarithmic averaging scale to be used, but when all last n values are expected to be kept then the history is also limited with n last time steps, older ones will be shifted out (last-in-first-out principle). The vector represents moving pattern of device service quality during its last n time steps. In case of logarithmic scale indicator element d_i is representing average over certain number of self-assessment values. It is important to guarantee that every self-assessment value during device lifetime is present in aggregated ways in some indicator element together with its neighbor values in time (continuity principle). This can be implemented in various ways e.g.:

- Linear scale (only last n values)
- Logarithmic scale
- Fibonacci scale
- Real-time scale (e.g. seconds, minutes, hours etc.)
- Real-time scale based on natural life-cycles of system from user point of view (e.g. seasons, years etc.)

Each scale has its own capacity. Linear scale of n elements has capacity n timesteps. Fibonacci scale - sum of n first Fibonacci numbers. Binary logarithm has capacity $2^n - 1$ timesteps. On Figure there is variation of binary logarithmic scale with some overlapping of vector data, limiting thus capacity of representation to $2^{n/2+1} - 2$ (i.e. in case of 10 elements history extends to 62 steps). To reduce the amount of memory historical data consumes, it is also possible to apply data compression methods that are generally used for signal compressing. Depending the importance of historical data, one could choose between lossy of lossless compression. Compression and decompression requires increased processing power but depending of the application it could be still more efficient rather than storing it to external storage or cloud. For experiments recursive Python script and graphic library of John M. Zelle [5] were used on Linux virtual machine. Self-assessment sequence was generated using random number generator, whereas frequency of "dangerous" values was artificially reduced - it is appearing only when two "dangerous" values are generated sequentially. It is expected to be close to behavior of real system in real environment - healthy device in normal working environment is expected

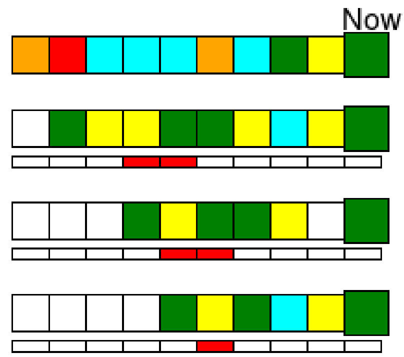

Fig. 5. Different lifelines after 32 steps. From top to down - linear (older steps lost), binary logarithm with overlap, Fibonacci and binary logarithmic. The rightmost element is the current value of self-assessment. Addition narrow scale represents dangerous/anomalous assessments met during respective averaged time-slot

to encounter rarely dangerous anomalies.

Pseudocode to calculate binary logarithmic lifeline, where pos is current position on lifeline and $status[]$ is integer array to represent status of specific lifeline element - empty (initial state), value written, or average over values written:

Algorithm 1 Binary logarithmic lifeline

```
procedure LIFELINEP2(g, pos = 0)
    if status[pos] = 0 then                          ▷ when empty
        line[pos] = g
        status[pos] = 1
        return
    else if status[pos] = 1 and pos ≥ 1 then
        line[pos] = (line[pos] + g)/2                ▷ averaging
        status[pos] = 2
        return
    else
        LIFELINEP2(line[pos], pos + 1)               ▷ recursion
        line[pos] = g
        status[pos] = 1
        return
    end if
end procedure
```

The problem with averaging is that the information about extreme cases will be finally lost. Average over long period of time will probably show satisfaction with service quality except when anomalous situation is truly persistent. To avoid that, in addition of averaging the ultimate negative assessment has to be presented together with average, e.g. the coloring of specific accompanying element red and/or counting negative assessments during the same interval. The interpretation of this count has to be done by user but existence of such information allows assess the system living environment and health - is it stable, improving or worsening. Logarithmic and Fibonacci scales are rather difficult to interpret by regular user i.e. they are not user friendly although mathematically simple and controller memory economic. Deciphering of visual feedback and understanding could be

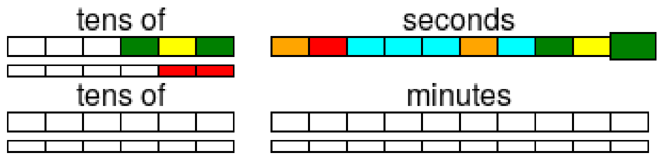

Fig. 6. Real-time clock based life-lines covering 32 steps of self-assessment. After every 10 seconds new element is added to "tens of seconds" register, etc.

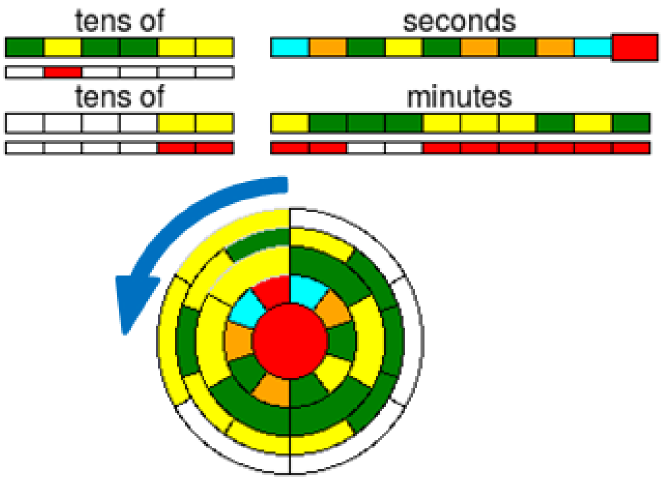

Fig. 7. Tree rings arrangement of real-time lifelines. After full turn of a ring average over ring elements is inserted to the first sector of next ring outward, rotating rings counter-clockwise. The instant "now" is in the center, following rings of seconds, tens of seconds, etc.

more straightforward when lifeline display is following some natural and hierarchical working cycles of device or user itself (real-time and calendar units).

Seeking history representation examples from nature, then tree rings on stump are source to assess fallen tree health history over its lifetime. Combing tree rings over longer period is possible to evaluate more global climate changes and natural disasters in past. It seems to be a good idea to arrange device self-assessment history into tree rings shape - far past in the middle and current instant on the outer ring (under the bark). Regrettably this is taking too much space on screen and ever-increasing area of outward expanding life-rings does not carry any additional information. Due to that quite an opposite representation could be proposed where current instant is in the middle and history is developing by ring sector outside. Figure 7 represents together with linear scale the same real-time clock-based life-meter in tree rings arrangement.

IV. Implementation

More feedback information means also more processing effort and components. If self-aware device has some sort of colour screen (e.g. LCD, OLED) then the visualization is just addition of an another SW module. The implementation of two-logarithm history function in SW or HW is rather simple because only addition and right shift (division by 2) are required. Required permanent storage (EEPROM) size is defined by double length of lifeline - one array for floats representing self-assessment value and respective array of integers to record number of anomalous situations. Both, lifeline and the number or rank of anomalous situations can be represented using arbitrary visualization symbols and colors. Self-aware devices without graphical or alphanumeric screen can be supplied with series of tri-color LED-s, which is good enough solution to express device service quality over limited number of time-steps. It is even possible to express information about dangerous anomaly situations alternating lifeline with some other anomaly count based color coding.

V. Conclusions and Future Work

In this work self-aware system lifeline indicator development issues were discussed and some possible lifetime service quality indicator designs were presented. The expected application area of such meters is wide, all contemporary IoTs and heterogeneous systems like cars, engines, computer components, home appliances, houses, etc. The best representation of self-assessment history is probably matter of design and research on base of the human perception properties. Although logarithmic scale lifeline can in principle be used to represent however long service history of the system then for realistic lifelines we suggest to use at least partially sub-logarithmic scale. I.e. most of systems do not last long enough to present meaningful averaged information over 10 years service interval 30 years ago. There is one more opportunity - if lifeline has some meaning to user then it has meaning to device itself too, thus that service quality history can be a source for correcting internal model of self-aware system. Anomalous situations registered past might happen cyclically again under the same environmental/seasonal conditions. I.e. longer pattern of lifeline might contain regular changes which can be learned to make better predictions about system capabilities in the future.

The self-assessment of service lifetime does not end at the terminal device. Assessment history can be uploaded to higher components in IoT hierarchy or to cloud for aggregated assessment over the system as a whole.

References

[1] L. Jiang and C. Yu, "Design and implementation of car black box based on embedded system," in *2010 International Conference on Electrical and Control Engineering*, pp. 3537–3539, IEEE, 2010.

[2] A. Jantsch and K. Tammemäe, "A framework of awareness for artificial subjects," in *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*, CODES '14, (New York, NY, USA), pp. 20:1–20:3, ACM, 2014.

[3] P. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao, "A survey of self-awareness and its application in computing systems," in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*, pp. 102–107, October 2011.

[4] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The Benefits of Self-Awareness and Attention in Fog and Mist Computing," *IEEE Computer*, vol. 48, pp. 37–45, July 2015.

[5] J. Zelle, *Python Programming: An Introduction to Computer Science*. Wilsonville: Franklin, Beedle & Associates Inc., 2003.