This coefficient of duty must be variable and specified by a software. Current strength of a signal must be 2 mA and the coefficient of duty must vary from 1/500 of seconds to 1/20 of seconds for generating correct signal.

Conclusion

Two main parts of the program for monitoring of recurrent laryngeal nerve by single-board computer Raspberry Pi model bare shown in the paper.

References

1. M. Dyvak, V. Tymets. "Emulation of programming environment for single-board computer Raspberry Pi at monitoring the recurrent laryngeal nerve", MEMSTECH, Lviv, pp.35-37, 2017
2. M. Dyvak et al., "Spectral analysis the information signal in the task of identification the recurrent laryngeal nerve in thyroid surgery" Przegląd Elektrotechniczny, vol. 89, no. 6, pp.275- 277, 2013.
3. Bush, Steve (25 May 2011). "Dongle computer lets kids discover programming on a TV". Electronics Weekly. Retrieved 11 July 2011.

UDC 004.4

NON-CLASSICAL PROPERTIES OF PROGRAM-ORIENTED LOGICS

Mykola Nikitchenko
Taras Shevchenko National University of Kyiv, DsC., professor

Software development is a grateful area of logic application. Logics can be used at every stage of development cycle, in particular, during requirement analysis, specification, design, verification, and testing.

To be successful, such logics should adequately represent essential features of the development stages. Among various logics, oriented on software development, the central place belongs to logics describing main features of computer programs. In particular, we identify the following features:

- *partiality*: programs may be undefined on some input data;
- *usage of complex system of data types*: arrays, hash-tables, linked lists, semi-structured data, etc.;
- *non-determinism*: programs can evaluate to different results on the same input data;
- *possibility of transformation* of abstract specifications (programs) to more concrete ones.

It is naturally to expect that implementation of these features in program logics will lead to substantial changes in reasoning rules. Even more, in some cases we will obtain non-classical reasoning rules.

In this paper we investigate non-classical properties of program-oriented logics. We restrict ourselves to considering the following program properties: 1) *partiality*, 2) *unfixed and unrestricted arity*, 3) *unessential variables*, 4) *sensitivity to unassigned variables*, and 5) *non-determinism*.

Let us discuss the impact of such properties on program-oriented logics [1, 2].

1. *Partiality* of program, functions, and predicates requires to change the consequence relation for the logic. Instead of classical relation which leads to totally-true predicates, we have to consider a dual consequence relation – the irrefutability relation – which leads to irrefutable predicates. As a result, we cannot rely on such traditional reasoning rules as modus ponens, cut, resolution etc.

2. *Unfixed and unrestricted arity*. In classical predicate logic we know precisely the number of variables upon which a predicate depends. This property permits to construct new predicates by substituting into initial predicate (into a formula) a fixed number of functions (terms). But if a predicate does not have fixed arity, we are obliged to change the definition of substitution which is formalized as superposition [1]. Such a superposition has different parameters indicating on function substituted into a predicate. This leads to new extended language of a logic [1, 2], and, as a result, to new language properties. If an arity is unrestricted we cannot say what variables are essential to a predicate and what are not. Therefore, those classical properties based on a known number of essential variables such as, say, coincidence lemma, properties of closed sentences, are not valid any more.

3. *Unessential variables*. In classical predicate logic it possible to identify free variables in a formula (which are essential variables), while other variables are unessential and can be used, for example, to substitute a quantified variables to obtain a formula variant. Such variants are necessary for different formula

transformations, in particular, to construct prenex normal forms. All such transformations rules are not valid if a logic does not have unessential variables.

4. *Sensitivity to unassigned variables.* Variable interpretations in classical predicate logic are assumed to be total. It means that every variable has a value in an interpretation. In this case Henkin axiom of the form $for\ all\ x\ P \rightarrow P$ is valid, but if we have interpretations with unassigned variables and predicates can detect such variables, then Henkin axiom fails.

5. In case of *non-determinism*, classical consequence relation collapses and we should use a special truth-false consequence relations [2, 3]. For these consequence relations many classical reasoning rules fail.

Summing up, we can say that many program features change classical rules for reasoning about programs, and thus, a special attention should be paid to identification of such situations and choice of special logics which will be valid in such cases.

References

1. M. Nikitchenko, S. Shkilniak. *Mathematical logic and theory of algorithms*. Publishing house of Taras Shevchenko National University of Kyiv, Kyiv, 2008 (in Ukrainian), 528 p.
2. M. Nikitchenko, S. Shkilnyak. *Applied Logic*, Publishing house of Taras Shevchenko National University of Kyiv, Kyiv, 2013 (in Ukrainian), 278 p.
3. A. Kryvolap, M. Nikitchenko, W. Schreiner, *Extending Floyd-Hoare logic for partial pre- and postconditions*, CCIS, vol. 412, Springer, Heidelberg, 2013, pp. 355-378.

UDC 005.8+331.45

STRUCTURE OF CONTEXT - SENSITIVE SOFTWARE SERVICE FOR NOTIFICATION ABOUT HUMAN SAFETY THREATS

Ruslan Shevchuk[1)], Anatolii Kliuiko[2)]

Ternopil National Economic University

[1)] Ph.D., associate professor, [2)] Master's Degree student

Introduction

It is known that human security depends on the surrounded objects and processes [1]. 60-80% of accidents are the results of the inability to predict, to identify hidden danger, to assess risk and align it with its capabilities, which are determined psycho-physiological properties of the body and state of the equipment [1]. Warning of threats related to the safety of human life is the urgent task. Developing of the context-dependent software services for personal mobile devices (PMD) is one way of solving this problem, which is a real-time report on possible threats or emergencies.

II. Formulation of the problem

Nowadays, there are a number of mobile applications developed for alerting and responding to threats to the security of human life: FamilyLocator [2] bSafe [3], SOS + [4] Shake2Safety [5] SaveMePro [6] "Cknopcka zizni" [7] "Mobilnyi spasatel "[8]. The principle of these applications is almost identical: they constantly monitor the whereabouts of the person and in case of threat (after pressing the panic button) it reports the persons coordinates by SMS or e-mail to the appropriate service or family. Some of these tools include first aid instruction in emergency situations and first aid facility guides [7,8].

The disadvantages of all analyzed services are static presentation of data and lack of information about the causes of threats. Accordingly, users who have received the alerts about the threat have no opportunities to adequately respond to alerts.

A number of analyzed applications are characterized by too congested interface in which the user in a state of shock, pain, confusion is not always able to select the required function [2,5,8].

The approach of the authors is to develop context-sensitive software service for permanent residence, which, in addition to the traditional functionality will be implemented with PMD user notification about potentially dangerous places to which the person is coming. Transfering the audio and video / photo from PMD sensor to data storage in the cloud in real time by pressing the panic button will be the unique service feature. In addition, the PMD's location and rout path will be displayed on a digital map.