Механика Транспорт Комуникации

ISSN 1312-3823 (print)
ISSN 2367-6620 (online)
том 14, брой 1, 2016 г.

Научно списание **http://www.mtc-aj.com** **статия № 1279**

СОФТУЕР ЗА КРИТИЧНИ ПО БЕЗОПАСНОСТ СИСТЕМИ – ПОНЯТИЯ И ПРИЛОЖЕНИЯ

Мария Христова
mhristova@vtu.bg

Висше транспортно училище „Тодор Каблешков”
София 1574, ул. „Гео Милев” № 158
БЪЛГАРИЯ

Ключови думи*: критични по безопасност системи, критичен по безопасност софтуер, грешки в софтуера, отказоустойчивост, надеждност, надеждност на софтуера*

Резюме: *Предмет на проучване е софтуерът, критичен за безопасността в различни приложения. Приведени са примери за компютърно базирани системи за управление в реално време от въздушния и железопътния транспорт, космоса, медицината, атомната енергетика и др., в които безопасността на човека зависи от правилното функциониране на системата.*

Целта на статията е да се структурира известната информация, свързана с понятийния апарат, използван в критични по безопасност системи и да се направи класификация на системите съобразно характера на управлявания технологичен процес. Представени са видовете откази в зависимост от техните последствия и изискванията към софтуера, произтичащи от стандартите за безопасност на системите.

1. ПОСТАНОВКА

Светът днес живее в информационно пространство, в което информационните и комуникационни технологии определят в решаваща степен параметрите на технологичните процеси, на инфраструктурата, на обществения живот и на безопасността на хората.

Предмет на проучване в този труд са проблемите на **софтуера**, **критичен за безопасността** в различни приложения.

Критични по безопасност системи (**S**afety **C**ritical **S**ystems - SCS) управляват особено отговорни технологични процеси, чиито откази могат да доведат до заплаха за здравето и/или загуба на живот, на големи материални и човешки ценности и/или недопустимо увреждане на околната среда. Неизпълнението на предвидените функции на SCS може да има животозастрашаващо въздействие.

Критична по безопасност система е система, при която безопасността на човека зависи от правилното функциониране на системата.

Безопасността *(safety)* е свойство на системата, което отразява способността ѝ да работи без заплаха за обкръжението на системата, без инциденти, без недопустим

риск, нараняване или смърт на човека. Безопасността не може да се гарантира абсолютно. Винаги остава някаква несигурност. Няма схема или система, която детерминистично да предотвратява инцидентите. Вероятността е подходящата мярка за изразяване на несигурността. В този смисъл винаги остава да съществува някаква опасност, която може да се оцени с вероятност. Въпросът е, дали тя е приемлива? Отговорът е свързан с риска.

Рискът е понятие, обединяващо вероятността за нежелано събитие с мащабите на предвидимите неблагоприятни последствия от него (заплаха за живота, имуществени щети, природни ценности и др.) [1]. Той може да е малък или голям, но има една негова стойност - ***граничен риск*** (приемлив, допустим, акцептиран), която определя дали една система е безопасна. Безопасността S(t) е отсъствие на недопустим на риск. Според стандарт MIL-STD-882D допустимостта се определя от горната граница на приемливото ниво на риска [2].

Целта на тази статия е да се проучи проблемното пространство на софтуера за релевантни по безопасност системи, за да се структурира известната информация, да се установят научните резултати, да се изведат и класифицират проблемите и се предложат насоки за решения на оставащите открити въпроси.

2. РЕЛЕВАНТНИ ПО БЕЗОПАСНОСТ СИСТЕМИ

Има много примери за особено отговорни технологични процеси и за релевантни системи с критичен по безопасност софтуер, които ги управляват.

В ***жп сигнализация, централизация и блокировки*** отказите в системите могат да предизвикат железопътни инциденти и катастрофи с тежки последици.

От изследванията и анализа на наличните данни в системите за управление на влаковете, използвани на борда на локомотивите, в железопътната и метро мрежа, се установява, че отказите в софтуера са били една от причините, които предизвикват срив на системата, включително с фатални последици [3]. Достатъчно е в програмата да остане неоткрита грешка, която води до управляващо въздействие за по-голяма от допустимата скорост или разрешение за движение, когато е забранено.

Във ***въздушния транспорт и космоса*** примерите са много**.** Това са преди всичко системите за управление на въздушното движение. Генерирани от софтуера данни се използват, за да се вземат решения, критични за безопасността на полетите [4]. Има над 15 000 пътнически самолети в света с общо около 18 милиона полета годишно. Средният процент на инциденти по софтуерни причини е около 1,4 на 1000000 кацания. В [5] се приема, че средната дължина на полета е от 5 часа, което прави норма на фатален инцидент на около 0.3. 10^{-6} 1/h. Изстрелването и провалът на *Ариана V* [6] и загубите на *Mars Polar Lander* и *Mars Climat Orbiter* [7] са свързани със софтуерни грешки. Има много примери за гибелта на изкуствени спътници или за катастрофални ситуации при използването на сложни динамични обекти като Марс-1 – 1976 г. и Аполо-13 - 1978 г. [8] както и във военни системи, които се дължат на относително прости грешки в софтуера.

В ***градския транспорт*** нараства сложността, обхватът и отговорностите на интелигентните транспортните системи за регулиране на трафичните потоци от един транспортен управляващ център (ТУЦ). Част от управляващите въздействия от ТУЦ са релевантни на безопасността. Те могат да предизвикат или най-малкото да създадат условия за инциденти.

В ***атомната енергетика*** днес в света има около 450 ядрени реактори, които се използват за производство на електроенергия в над 30 страни по целия свят. В [9] се дава информация за сложна софтуерна програма в релевантна по безопасност система

за управление на атомна централа с повече от 650 микропроцесори и 1200 платки и с над 70000 линии за комуникация. С развитие на технологиите безопасността и ефективността на реактора нараства. Предлагат се нови по-безопасни (но обикновено неизпитани) реактори. Няма обаче гаранция, че реакторите ще бъдат проектирани, програмирани, изградени и експлоатирани безгрешно. Проектираните, например във Фукушима, реактори не са съобразени с цунами с такава сила, каквито се случиха през 2010 г. Интердисциплинарен екип от Масачузетския технологичен институт (MIT, Бостън) е изчислил, че като се отчете очакваният растеж на ядрената енергия до 2055 г., може да се очакват най-малко четири сериозни ядрени аварии, част от които могат да са причинени от грешки в софтуера.

В ***медицината*** има редица примери за приложение на системи и устройства с животоосигуряващо значение. Такива са системите за лъчева терапия. Такива са критичните медицински данни (за медицинското състояние, медицинска документация, кръвни банкови данни и др.), съхранявани в компютърната памет. Такава е техниката кардиопулмонарен байпас, която временно поема функцията на сърцето и белите дробове по време на операция и поддържа циркулацията на кръвта и съдържанието на кислород в тялото. Към критичните системи се отнася и инсулиновата помпа, която доставя правилното (според текущото ниво на кръвна захар) количество инсулин, като най-важното е да не се допусне свръхдоза, опасна за живота. Широкото използване на компютри в хирургични процедури - операция на гръбначния стълб, очната хирургия, използването на компютърно контролирани роботизирани устройства вместо традиционни хирургични инструменти и т.н - има голямо бъдеще.

Телемедицината в реално време [10, 17] също е критична по безопасност система. Вместо хоспитализация, в много случаи ще бъде възможно хората да си стоят в къщи, но да бъдат наблюдавани от разстояние и лекувани чрез автоматизирано оборудване в дома.

Без да се навлиза в примери и детайли, могат да се маркират релевантни по безопасност системи още в комуникациите, в банковото дело, в автомобилите, в охранителните и алармени системи, в строителството и др.

3. ПРИЧИНИ ЗА ОТКАЗИ, ФУНКЦИОНАЛНИ ОТКАЗИ, ПОСЛЕДСТВИЯ ОТ ОТКАЗИТЕ

3.1 Таксономия

Надеждността е *вероятност, че софтуерът ще изпълнява своята функция по предназначение в предвиденото време при установени условия на околната среда.*

Надеждността на софтуера е *свойството на съставящите го програми да изпълняват зададените функции в зададените условия на конкретното компютърно техническо средство.* Съставни понятия на надеждността са **отказът**, **работоспособността**, **грешките** и **повредите**.

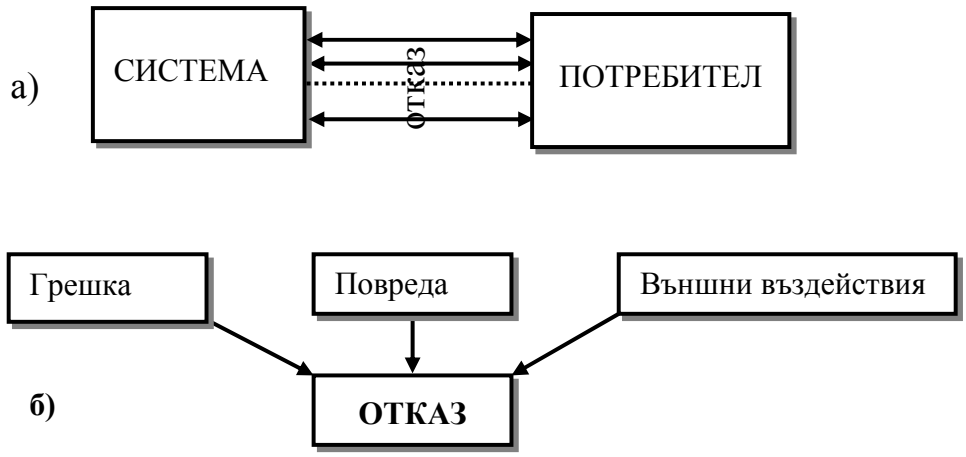
Функционалният отказ (*Failure*) е събитие, вследствие на което обектът (система, устройство, възел, програма, агрегат и т.н.) излиза от работоспособност [11]. Проявява се на граничната плоскост между системата и нейния потребител (фиг.1а).

Работоспособността е състояние, в което *параметрите, характеризиращи **съществените** свойства на обекта, са в допустимите граници, определени от спецификацията и нормативно-техническата документация.* Съществени са свойствата, от *които зависи използването на обекта за целите на предназначението му* [12]*, т.е., които определят неговата функционалност.*

Причините за отказ могат да се класифицират както следва:

1. ***Непреднамерени***

- *Случайни субективни причини:* неоткрити и/или неотстранени ***грешки на софтуера (Errors*)**: конструкторски, технологични, проектантски и други грешки; операторски грешки - диспечерски, на водачи, на машинисти и др.; грешки на поддържащия персонал по време на експлоатацията (в техническото обслужване, ремонта и възстановяването).



фиг. 1 Взаимоотношение отказ – причини за отказ

- *Случайни обективни причини* ***(Fault)*** *по време на експлоатация на системата***:** повреди в резултат от случайни обективни въздействия (остаряване, износване, умора, разрегулиране, изхабяване и пр.), дефекти в градивните компоненти, останали след пускането в експлоатация и др.
- *Външни обективни въздействия:* нежелани индуктивни въздействия, влияния на електрозахранването на други технически системи, метеорологични шумове, промишлени смущения, влияние на електротракцията в транспорта и пр.

2. ***Преднамерени* (*злоумишлени намеси*)**

- *Злонамерени нарушители на функционирането на системите*.
- *Кракерски атаки*.

Злонамерените намеси са свързани със сигурността. **Сигурността** е защитата срещу злонамерени действия. Тя не е предмет на настоящото проучване.

В зависимост от компонентите, в които могат да се случат, причините за откази са (фиг.1,б):

- В ***софтуера.*** Грешки могат да бъдат допуснати на всички етапи от жизнения цикъл на програмното осигуряване. Ако спецификацията не е правилна, системата може да ѝ съответства, т.е., да няма грешки, но да предизвика инцидент. Повечето софтуерни провали са резултат от дефекти на заданието, а не дефектни изчисления. Това значи, че софтуерните грешки могат да са и концептуални, те могат да се намират на най-високото ниво. Софтуерните грешки са атрибут на всички устройства, работещи с един и същ софтуер.
- В ***хардуера.*** Те могат да се дължат на производствени дефекти или да възникват по време на работа при продължителна експлоатация. Водят до случайни повреди или кратковременна самоотстраняваща се загуба на работоспособност. Появяват се като неизправности, които нарушават някои от параметрите и характеристиките, съдържащи се в спецификацията и нормативно-техническата

документация. Макар и произведени по същата технология и документация и работещи по същата програма, хардуерните неизправности се свързват само с конкретното устройство и са независими от повредите на другите устройства от същата серия. Това е тяхна съществена характеристика.

- В ***комуникациите***. Едно съобщение може да се трансформира в друго, което да има по-висока отговорност за безопасността [13]. Изкривяването на сигнала е допустимо само в обратната посока. Има и контекстно-зависими команди, напр. подаване на вярна команда, но в грешно време.
- По вина на ***човеко-машинния интерфейс*** (MMI) на системата. Хората правят грешки. Грешките в MMI най-често са основни причини за системни откази. Отказите на MMI преобладаващо са резултат от операторска грешка, която е допусната при изключено автоматично управление [14].

В контекста на този труд особено значение има понятието ***програмна грешка***. Грешките влияят и на надеждността, и на безопасността. Надеждността на софтуера може да се подобри чрез отстраняване на грешки, но отстранените грешки могат и да не засягат безопасността. Ако между тях няма такива, които водят до опасни откази, безопасността не се подобрява. Софтуерът може да предизвика катастрофални откази, макар че работи както е предвидено. Най-критичните за безопасността са софтуерни грешки поради неправилно поставени изисквания в спецификацията. Предвид особеното значение на грешките в софтуера, критичен за безопасността, част от проблемите на критичния софтуер са свързани с тяхната характеристика, с методите за тяхното откриване, както и с количествените оценки за влиянието им върху безопасността и надеждността.

4. ДВА КЛАСА SCS

Нарушаване на правилното функциониране на системите, свързани с безопасността, може да настъпи поради откази. В зависимост от характера на особено отговорните технологични процеси, които управляват, техническите решения се подразделят на два основни класа [15], от които зависи и характера на отказите.

Към първия клас се отнасят ***SCTP*** **(S**pecial **C**ritical **T**echnology **P**rocesses), които са подходящи да се дефинира критерий за безопасно поведение след отказ на управляващите ги системи. Това е критерий, по който се ограничава функционалността или се спира управляваният процес [16]. Ако отказът противоречи на критерия, той е ***опасен*** (*hazard*)**.** Ако се съгласува с критерия, е ***защитен*** (*safety*). Защитните откази водят до нежелани, но неопасни прекъсвания на SCTP, чрез които се създава принуда за отстраняване на отказа, за да може процесът да продължи. За такива системи са казва, че имат **fail-safe** поведение.

Към втория клас се отнасят системи**,** в които *fail-safe* поведение е нецелесъобразно: във въздушния транспорт, най-често в медицината, в животоосигуряващите системи и др. всяко спиране на процеса е недопустимо. Естеството на процеса е такова, че не може да се дефинира критерий за безопасно след отказ поведение. Към подобни SCS се поставят изисквания за готовност или непрекъсваемост на SCTP. И в тези случаи се говори за SCS, но те вече не са *fail-safe.*

Има по-общ критерий, по който системите попадат в класа на SCS и от двете групи. Той е допустимостта на риска, който произтича от евентуален отказ, нормиран със стандарти за съответния SCS-клас. Това са стандарти за безопасност, които са общовалидни, нерелевантни към техническото решение и се отнасят и за двете разгледани групи.

Ако нормата на риска се постига и чрез висока надеждност, не е нужно да се дефинира поведение. Тя може да се постигне и чрез ***отказоустойчивост*** (***fault-***

tolerance), която неминуемо изиска някакъв по-голям ***излишък*** (***redundancy***) и свързани с него ресурси.

ЗАКЛЮЧЕНИЕ

Направени са проучвания на литературните източници, свързани със софтуера за критични по безопасност системи. Установява се, че макар и в различна степен зависими от безопасността, системите за управление на технологичните процеси, от чието правилно функциониране зависи риска за човека, имат силно присъствие в инфраструктурата и тяхното приложение все повече нараства. С това и проблемите на софтуера за критични приложения придобива все по-голяма значимост.

Настоящата статия отразява част от проучванията на проблемното пространство, които по-нататък извеждат обобщенията, че остават непреходно актуални редица открити въпроси, заслужаващи вниманието на науката.

ЛИТЕРАТУРА

[1] Sommerville I., Safety Engineering, pp. 1-36, 2013

[2] Lyu M. R., Handbook of Software Reliability Engineering, McGraw-Hill, 1996

[3] Caia H., Ch. Zhanga, W. Wub, T. Hoa, Z. Zhangb, Modelling High Integrity Transport Systems by Formal Methods, International Conference on Traffic & Transportation Studies (ICTTS'2014) 729 - 737, 2014

[4] Bérard B. et al., Systems and Software Verification - Model-Checking Techniques and Tools, Springer Verlag, 2001

[5] Shooman M. L., Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design, A Wiley-Interscience Publication, John Wiley$ Sons, inc., ISBN: 978-0-471-29342-2, 2002

[6] Jezequel J.M., B. Meyer, Put it in the Contract: The Lessons of Ariane, Computer, Vol. 30, No. 2, 1997, pp.129-130

[7] Аджиев В., Мифы о безопасном ПО: уроки знаменитых катастроф, Открытые системы, 06, 1998

[8] Butler, R.W., G.B. Finelli, The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software, IEEE Transactions on Software Engineering, 19(1), pp. 3-12, 2001

[9] Knight J. C., Safety critical systems: challenges and directions, Proceedings ICSE '02, pp 547-550, ACM, NY, USA ©2002

[10] Norris, A. C. (2002). Essentials of Telemedicine and Telecare. West Sussex, England; New York: John Wiley & Sons, Ltd. ISBN 0-471-53151-0.

[11] Христов Х., В. Трифонов, Надеждност и сигурност на телекомуникациите, Издателство Нови знания, 2007

[12] M. Franeková, P. Lüley**,** ModelLing of failure effects within safety- related communications with safety code for railway applications. Journal Mechanic, Transport, Communication, ISSN 1312-3823, art. ID 1213, VII 27 –VII 34, 2015

[13] Hristov H., W. Bo, Safety Critical Computer Systems: failure independence and software diversity effects on reliability of dual channel structures, Information Technologies and Control, № 2, pp. 9-18, 2014

[14] Христов Хр., М. Христова, Н. Георгиев, Аналитичен подход и модел за анализ и оценка на безопасността на човекомашинни системи за управление на експлоатационния процес в транспорта, Научно списание „Механика, Транспорт, Комуникации", ISSN 1312-3823, статия 0480, BG–3.17 –BG-3.28, бр. 2, 2010 г.

15] Ahmed F., Analysis of requirements and communication in safety-critical software systems. https://cs.uwaterloo.ca/~dberry/ATRE/Slides/Ahmed.pdf
[16] Elakeili, S. M., Fail-Safe Test Generation of Safety Critical Systems Electronic Theses and Dissertations,Paper 180, 2015
[17] Popov G., Ianchev G., Krasteva A., Temedicine - possibilities and development in Bulgaria, International Scientific Conference "CompSystTech 2004", Sofia, dec., 2004, pp.245-247

SOFTWARE FOR SAFETY CRITICAL SYSTEMS - CONCEPTS AND APPLICATIONS

Mariya Hristova
mhristova@vtu.bg

Todor Kableshkov University of Transport
1574 Sofia, Geo Milev str. 158
BULGARIA

Key words: *safety critical systems, safety critical software, errors in software, fault tolerance, reliability, reliability software*

Abstract*: The paper is dedicated to a study on the problems of safety critical software in various applications. The examples of computer-based systems for real-time management have been taken from air and rail transport, space, medicine, nuclear power engineering, etc. where human safety depends on proper functioning of the system.*

The aim is to structure the existing known information related to the conceptual apparatus used in safety critical systems, classification of systems accordance with the nature of the controlled process, the types of failures according to their consequences and A software requirements specification arising from the safety standards of the systems.