

AXBOROT TIZIMLAR SIFATINI TAMINLASH YONDASHUVLARI VA USULLARI

t.f.d., professor ¹Mo'minov Baxodir Baltayevich

²Muxamadiyev Sanjar Isoyevich

<https://www.doi.org/10.5281/zenodo.8062728>

ARTICLE INFO

Received: 12th June 2023

Accepted: 19th June 2023

Online: 21th June 2023

KEY WORDS

Axborot tizimi, Axborot tizimining sifat ko'rsatgichlari, validatsiya, verifikatsiya, statik tahlil, formal usul, dinamik usul, sintetik usul mantiqiy-algebraik model, bajariladigan model, funkcionallik, ishonchlilik, foydalanishga qulaylik, samaradorlik, xizmat ko'rsatish qulayligi, portativlik.

ABSTRACT

Axborot tizimlariga bo'lgan talabning oshib borishi axborot tizimlarini jadal suratlarda ishlab chiqarilishiga va ko'payib borishiga olib kelmoqda. Maqolada ularning to'g'riligi va sifatini tekshirishda validatsiya va verifikatsiya usullari muhokama qilingan. Axborot tizimlarining sifat ko'rsatgichlari ISO hamda IEEE standartlariga ko'ra qanday parametrlar yordamida aniqlanishi keltirilgan. Verifikatsiya va validatsiya tushunchalarining farqli va umumiy jihatlari yoritib berilgan. Verifikatsiya usullarining qo'llanilish sohalari va beradigan natijalari tahlil qilingan. Verifikatsiya usullaridan foydalanib axborot tizimlarini tekshirish instrumentlari keltirib o'tilgan. Zamonaviy axborot tizimlari sifatini tekshirish va baholash uchun ahamiyatga ega parametrlar, artefaktlar muhokama qilingan.

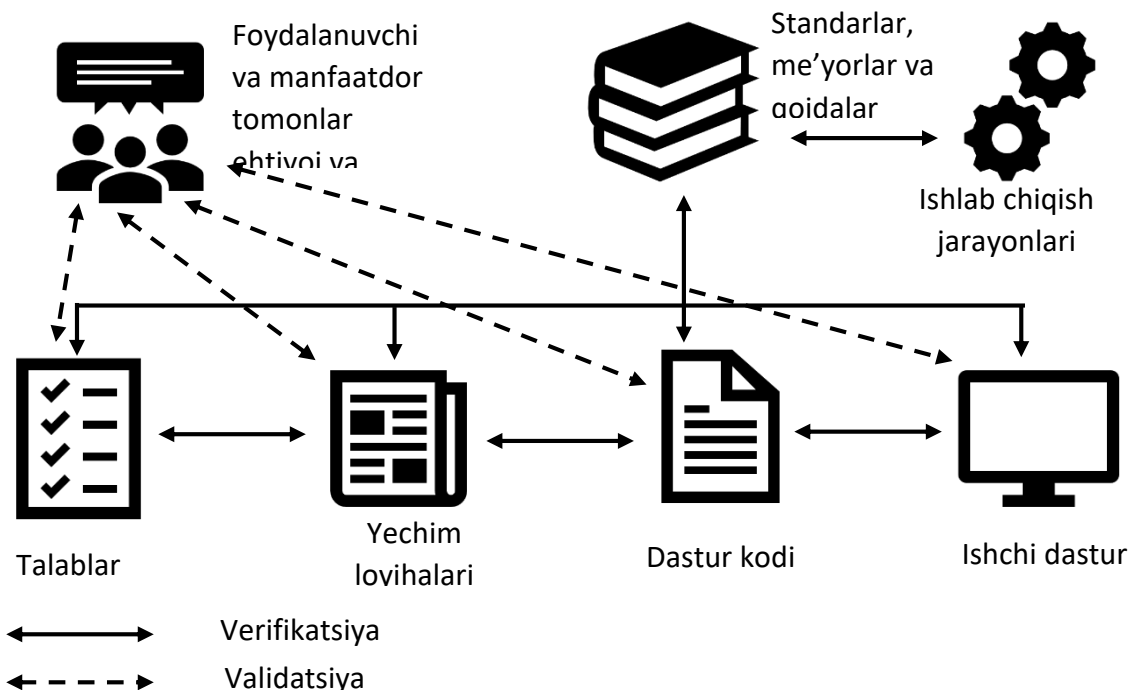
Kirish

Axborot texnologiyalarining rivojlanib borishi inson faoliyatining barcha sohalariga axborot texnologiyalarini qo'llab ish samaradorligi, aniqligi va ishonchliligini oshirish imkoniyatini taqdim etib kelmoqda. Axborot texnologiyalaridan to'g'ri va samarali foydalanish iqtisodiyot sohalarini rivojlantirishning asosiy omili sifatida o'zini namoyon qilib kelmoqda. Axborot texnologiyalari deganda raqamli qurilmalar va ularning ishini boshqaruvchi dasturiy vositalar va axborotlarga ishlov beruvchi algoritm hamda usullar tushuniladi. Iqtisodiyot sohalarini avtomatlashtirish axborot tizimlari yordamida amalga oshiriladi. Shuning hisobidan axborot tizimlarida bo'lgan talab kundan kunga ortib bormoqda. Axborot tizimlariga bo'lgan talabning ortishi murakkab jarayonlarni ham avtomatlashtirishni talab etmoqda. Bu esa o'z-o'zidan axborot tizimlarining murakkablashishiga, hajm va ko'lam jihatdan o'sib borishiga sabab bo'lmoqda. Ko'pchilik olimlar zamonaviy dasturiy komplekslar insonlar tomonidan yaratilgan eng murakkab tizimlar deb hisoblashadi. Tizimlarning murakkablashishi xatolarning ko'payishiga va xatolar sabab zararlarning o'sishiga olib keladi.

Verifikatsiya va validatsiyadan foydalanilgan axborot tizimlarining to'g'riligini, sifatini tekshirish va undagi xatolarni aniqlashga mumkin. Bularning maqsadi bir xil bo'lsada ular tekshiradigan xususiyatlar, qoidalar va cheklovlar manbalari bilan farqlanadi.

Verifikatsiyaning maqsadi axborot tizimiga qo'yilgan talablarga javob berishini va mo'ljallangan muhitda ishonchli va samarali ishlashini ta'minlash orqali uning sifatini yaxshilashdir. Axborot tizimi artefaktlari va jarayonlarini o'rnatilgan me'yorlar va standartlarga muvofiqligini tekshirish orqali ishlab chiqaruvchilar yuqori sifatni hamda mijoz va foydalanuvchilar talablarini qondirishlari mumkin. Verifikatsiya qo'lda yoki kodni ko'rib chiqish, statik tahlil, test va modelni tekshirish kabi avtomatlashtirilgan vositalar va usullar yordamida amalga oshirilishi mumkin. Verifikatsiya jarayoni odatda talablar hujjatlarida, loyiha spetsifikatsiyalarida, kodlash standartlarida va boshqa ma'lumotnoma materiallarida ko'rsatilganidek, dasturiy artefaktlarni kutilgan hatti-harakatlari va xususiyatlari bilan taqqoslashni o'z ichiga oladi. Ushbu artefaktlar orasidagi har qanday nomuvofiqliklar yoki qarama-qarshiliklar nuqson(xato)lar sifatida aniqlab olinadi va keyinchalik ular ishlab chiquvchilar tomonidan tuzatiladi. Bunday tekshiruvlarning o'tkazishdan asosiy maqsad keying bosqichlarda tuzatish qiyin va qimmat bo'ladigan xato va nuqsonlarni aniqlash va ularni tuzatish hisoblanadi.

Validatsiya - bu dasturiy mahsulotning maqsadli foydalanish kontekstida uning mo'ljallangan foydalanuvchilari va manfaatdor tomonlarning ehtiyojlari va talablariga javob berishini baholash jarayoni. Validatsiyaning maqsadi dasturiy mahsulot mijozning talablarini qondirishini va mo'ljallangan iqtisodiy samaradorlikni berishini ta'minlashdir. Validatsiya odatda foydalanuvchilar, mijozlar, mavzu bo'yicha ekspertlar va dasturiy mahsulotning haqiqiy ehtiyojlari va talablari haqida tushuncha beradigan boshqa manfaatdor tomonlar bilan hamkorlikda amalga oshiriladi[14].



1-rasm. Verifikatsiya va validatsiya tushunchalarining farqli va umumiy jihatlari.

Axborot tizimlarining sifat ko'rsatgichlari va ularni tekshirishda foydalaniladigan usullar.

Ishlab chiqilgan axborot tizimining sifatini ta'minlash tizimdan foydalanish samaradorligiga, unumdorligiga va kelgusidagi istiqboliga bevosita ta'sir qiladi. Axborot



tizimining sifatini baholashning xalqaro va milliy standartlari va me'zonlari mavjud. ular tizimning sifatini nazorat qilishda foydalaniladi. ishlatiladigan verifikatsiya va validatsiya mavjud. qo'yilgan maqsadga mosligi, foydalanuvchilarga qulayligini, tizimni yaratishdan manfaatdor tomonlarning talablarini qondirish darajasini hamda xavfsizlik



2-rasm. ISO 9126[4] standartida axborot tizimlarining sifat ko'rsatgichlari

Funktsionallik tizimning foydalanuvchiga zarur bo'lgan vazifalarni to'g'ri bajarish, standartlarga mos qiymatlar qaytarish hamda tizimdan xavfsiz foydalanish qobiliyatini bildiradi.

Ishonchlilik-ma'lum vaziyatlarda o'z ishchanlik qobiliyatini saqlab qolish ya'ni xatolik va uzilishlarga bardoshliliigi hamda amallar bajarishdagi muvaffaqiyatsizliklardan keyin ma'lum vaqt ichida ishchanlik qobiliyatini va ma'lumotlar yaxlitligini saqlab qolishidir.

Foydalanishga qulayligi yoki amaliyligi – foydalanuvchilarga tushunarli, foydalanishga qulay, o'rganish oson, dizayn jihatdan jozibador va standartlarga mos bo'lishidir.

Samaradorlik yoki ish unumdorligi-berilgan vazifani bajarishda sarflanadigan resurslardan(vaqt, xotira, trafik va h.k) foydalanish samaradorligi va unumdorlik standartlarga mosligi.

Xizmat ko'rsatish qulayligi-tizimni rivojlantirish va qo'llab quvvatlash bilan bog'liq ishlarni(tahlil qilish, o'zgartirishlar kiritish va tekshirish) amalga oshirish imkoniyati va osonligi hamda ularga barqarorligi.

Portativligi yoko mobilligi -bir muhitdan boshqa muhitga(tashkiliy, qurilmaviy va dasturiy) o'tkazilganda o'z ishlash qobiliyatini saqlab qolishi, ya'ni moslashuvchanligi, o'rnatish qulayligi va birgalikda ishlab ketish qobiliyati.

Axborot tizimini verifikatsiya va validatsiya qilish jarayonlarni rejalashtirish va o'tkazish bo'yicha IEEE 1012 standarti mavjud bo'lib unda ISO 12207 da belgilangan jarayonlar doirasida turli faoliyat sohalariga mos keladigan verifikatsiya masalalari to'plamining



tavsiflari keltirilgan. Unda verifikatsiya jarayonlarini turli artefaktlar bo'yicha amalga oshirishni rejalashtirish, o'tkazish va talablarga mosligini baholash yoritib berilgan. IEEE 829 standarti axborot tizimlarini testlash amalga oshirish tartiblarini o'z ichiga oladi. Axborot tizimlarini baholash va ularni tekshirish usullari uchun sanab o'tilgan standartlardan boshqa standartlar ham mavjud.

Verifikatsiya metodlarini sinflarga ajratib ko'rsatgan. Unga ko'ra ekspertiza, statik tahlil, formal usullar, dinamik usullar va sintetik usullar mavjud. Ekspertiza usullari Fagan, Gilb, Graham, Bisant, Lyle, Uchrashuvlarsiz baholash, Loyihani faol baholash (active design review), Britcher, Bosqichli tekshiruvlar, N katlamli baholash [6] usullari kengroq tarqalgan. Ekspertiza usullaridan axborot tizimi arxitekturasini tahlil qilishning eng mashhurlari SAAM (Software Architecture Analysis Method) [9], ATAM (Architecture Tradeoff Analysis Method) [12], SBAR (Scenario-Based Architecture Reengineering), ALMA (Architecture-Level Modifiability Analysis) [11], CBAM (Cost Benefit Analysis Method) usullaridir. Bu usullarning amalga oshirish instrumentlari juda kam va to'liq emas.

Statik tahlil usullari artefaktlarini to'g'ri shakllantirilganligi, rasmiylashtirish qoidalarga rioya qilinganligini tekshirish va ulardagi xato va kamchiliklarni ma'lum shablonlar asosida izlash orqali amalga oshiriladi [13]. Statik tahlil usullari dastur manba kodini tekshirishda ko'proq qo'llaniladi [2]. Shablon asosida xatolarni aniqlash kompilyatorlarda semantik tahlil uchun juda samarali qo'llaniladi. Ayrim modellarni tekshirishda ham ushbu usuldan foydalanish mumkin [3].

Formal usullarda axborot tizimi xususiyatlarini tahlil qilish uchun axborot tizimi modellaridan foydalaniladi. Mantiqiy-algebraik modellar axborot tizimi xususiyatlarining ma'lum bir to'plamini tavsiflaydi, vaqt o'tishi bilan o'zgarishi mumkin, lekin bu xususiyatlar qanday o'zgarishi haqida ahamiyatga ega emas. Mantiqiy modellarni ifodalashda mulohazalar hisobi, predikatlar hisobi, yuqori tartibli predikatlar hisobi, λ -hisobi, modal mantiq, vaqt mantig'i, chiziqli vaqt mantig'i, μ -hisobi, aniq vaqt mantig'i kabi hisoblashlardan foydalaniladi. Algebraik modellarni ifodalashda relyatsion(munosabat) algebraisi, ma'lumotlarning abstrakt tiplari algebraik modeli, algebraik jarayonlardan foydalaniladi [5].

Bajariladigan yoki operativ modellari (executable models) modellashtiriladigan dasturiy ta'minot xususiyatlarining o'zgarishini kuzatish uchun bajarilishi mumkin bo'lgan model. Bu modellarga chekli avtomatlar, belgilangan o'tish tizimlari (labeled transition systems), kengaytirilgan chekli holat mashinalari (extended finite state machines), ta'sirlashuvchi avtomatlar (communicating finite state machines, CFSM), ierarxik avtomatlar, vaqt avtomatlari, gibrid avtomatlar, Petri to'ri [1,7], ω -avtomati, abstrakt (mavhum) holat mashinalari va shu kabilar misol bo'ladi.

Oraliq modellar ham mantiqiy-algebraik, ham bajariladigan xususiyatlarga ega bo'lgan modellar hisoblanadi. Yuqorida sanab o'tilgan modellarning ba'zilar ikkala sinfga ham tegishli bo'lishi mumkin. Biroq, mantiqiy-algebraik va bajariladigan elementlar to'g'ridan-to'g'ri birlashtirilgan modellarning turlari mavjud. Ularning asosiy turlari Hoare mantiqlari, dinamik yoki dastur mantig'i (dynamic logics, program logics), dasturiy ta'minot shartnomalari hisoblanadi.

Formal usullar yordamida ma'lum xususiyatlarning ishlashini tekshirish uchun xususiyatlarni va tekshirilayotgan artefaktni formallashtirish kerak, ya'ni ikkalasi uchun



formal model yaratish kerak. Tekshirish uchun xususiyatlar modeli odatda spetsifikatsiya va tekshirilayotgan artefakt modeli realizatsiya deb ataladi. Agar spetsifikatsiya S va realizatsiya I bo'lsa hamda ularning ikkalasi ham mantiqiy-algebraik model bo'lsa $I \vdash S$ deb yoziladi. Bunday tekshirishni amalga oshirishda deduktiv tahlil usuli qo'llaniladi, ya'ni biror spetsifikatsiyani ifodalovchi mulohazalar to'plamini tekshirish orqali amalga oshiriladi. Deduktiv tahlil dastlab Floyd va Hoare tomonidan taklif qilingan usullar yordamida amalga oshirilgan. Ushbu usullar Hoare mantig'iga va Floyd tomonidan taklif qilingan tsiklning invariantlariga va uni bajarish jarayonida monoton ravishda o'zgarib turadigan baholash funktsiyalariga $\{\Phi_i \ \& \ \phi = a\}$ $P_{ij} \ \{\Phi_j \ \& \ \phi < a\}$ asoslangan. Buni soddalashtirish uchun Dijkstra zaif old shartlarni qurish texnikasidan foydalanishni taklif qildi. $\Phi \Rightarrow wp(P, \Psi)$. Deduktiv tahlilni inson amalga oshirishi mumkin, biroq avtomatik ravishda spetsifikatsiya va realizatsiya uchun dalillarni yaratish vositalaidan foydalanish mumkin. Deduktiv tahlil instrumentlari kengaytirilgan taklif mantig'I yoki birinchi tartibli mantiqqa asoslangan (ACL2, E-SETHEO, Key, Vampire, Waldmeister, Darwin va h.k), yuqori tartibli mantiqqa asoslangan (PVS, HOL, Isabelle, Coq va h.k) bo'ladi[34].

Agar S mantiqiy-algebraik va I bajariladigan model bo'lsa $I \models S$ deb yoziladi. Bu model tekshiruvda asosan Model Checking (modelni tekshirish) usullaridan foydalaniladi va bu ishlar inson tomonidan emas balki maxsus vosita yordamida amalga oshiriladi. Bu usullar yordamida axborot tizimi xususiyatlaridan xavfsizlik, yashovchanlik, turg'unlik va to'g'rilik tekshirilishi mumkin. Model checking usullarida Kripke modelidan, LTL va CTL mantiqlaridan, OBDD diagrammasidan, μ - hisoblashlari algoritmlaridan foydalaniladi. Axborot tizimlarini verifikatsiya qilishda foydalaniladigan modelni tekshirishda SPIN, Evaluator, SMV, NuSMV, Design/CPN, UPPAAL, Kronos, Bogor, HyTech instrumentlaridan ko'proq foydalaniladi.

Agar S va I bajariladigan model bo'lsa umumiy qabul qilingan nom yoki belgilanish yo'q biroq biz uni muvofiqlikni tekshirish deb yuritsak bo'ladi. Muvofiqlikni tekshirish tekshirilishi kerak bo'lgan xususiyatlar umumlashtirilgan avtomat (o'tish tizimlari, Petri to'ri va boshqalar) ko'rinishida taqdim etilgan tizim yoki uning komponentining xatti-harakatlariga qo'yiladigan talablar bo'lishi mumkin. Muvofiqlikni tekshirishda Verity-Check, Bisimulator, Reductor instrumentlaridan ko'proq foydalaniladi.

Dinamik verifikatsiya usullari tayyor axborot tizimlari sifatini tizim natijalarining talablardagi natijalarga va loyiha qarorlariga muvofiqligini tekshirish uchun foydalanadi. Dinamik verifikatsiya xizmat ko'rsatish qulayligidan boshqa barcha sifat ko'rsatgichlarini tekshirishi mumkin. Dinamik verifikatsiya testlash va monitoring orqali amalga oshiriladi. Testlash va monitoring qiluvchi instrumentlar mavjud.

Sintetik(gibrid) verifikatsiya usullari bu bir nechta verifikatsiya usullaridan birgalikda foydalaniladigan usullar bo'lib hisoblanadi. Ushbu usullarning ba'zilar so'nggi 10-15 yil ichida mustaqil, tez rivojlanayotgan tadqiqot yo'nalishlarini, modelga asoslangan test va formal xususiyatlarning monitoringini keltirib chiqardi[14].

Xulosa va takliflar

Xulosa qilib aytganda, verifikatsiya va validatsiya dasturiy ta'minot injiniringida dasturiy mahsulotlarning sifati va samaradorligini ta'minlashga yordam beradigan ikkita muhim faoliyatdir. verifikatsiya dasturiy artefaktlarni spetsifikatsiyalar va standartlarga muvofiq



baholashga qaratilgan bo'lsa, validatsiya dasturiy mahsulotlarni ulardan maqsadli foydalanish kontekstida foydalanuvchilar va manfaatdor tomonlarning ehtiyojlari va kutishlariga muvofiq tekshirishga qaratilgan.

Verifikatsiya metodlaridan axborot tizimlarini ishlab chiqish jarayonida foydalanish axborot tizim sifatini oshiradi. Axborot tizimlariga bo'lgan talablarning takomillashib borishi ularning sifatini tekshirish va baholashda har bir artefaktning alohida tekshirish zaruratini keltirib chiqarmoqda. Yuqorida tahlil qilingan verifikatsiya usullarining ko'pchiligi dastur kodining, yechim modelining to'g'riligini, berilgan tetslarda tizimning to'g'ri ishlashini, amallar ketma ketligining to'g'riligini tekshirishga mo'ljallangan bo'lib hisoblanadi. Axborot tizimining integratsiya imkoniyatlarini, moslashuvchanligini, reinjinerining qobiliyatini tekshirish uchun verifikatsiya usullari mavjud emas. Vaholangki bugungi kunda tizim reinjineringga, moslashuvchanligi va integratsiya qobiliyati axborot tizimining ta'minlanishi zarur bo'lgan sifat ko'rsatgichlari hisoblanadi.

References:

1. Mo'minov, B.B., Eshankulov, K. Modelling asynchronous parallel process with Petri net. //International Journal of Engineering and Advanced Technology 8(5 Special Issue 3), c. 400-405. 2019.
2. Мўминов Б.Б., Мухаммадиев С.И. Дастурларни синтактик таҳлил қилишда ҳосил қилувчи грамматикалар усули// Муҳаммад ал-Хоразмий авлодлари– 2021. – №4(18). – 3-8 бетлар..
3. Mo'minov B.; Dauletov A.; Classes of electronic document circulation systems and mathematical models of processing, Central Asian Journal of Education and Computer Sciences (CAJECS), 1, 2, 6-16, 2022.
4. [4] ISO/IEC 9126-1 Software engineering – Product quality. Geneva, Switzerland: ISO, 2004.
5. T. Gilb, D. Graham. Software Inspection. Addison-Wesley, 1993.
6. C. Kaner. The Performance of the N-Fold Requirement Inspection Method. Requirements Engineering Journal, 2(2):114-116, 1998.
7. G. M. Schneider, J. Martin, W. T. Tsai. An experimental study of fault detection in user requirements documents. ACM Transactions on Software Engineering and Methodology, 1(2):188-204, 1992.
8. Boltaevich, M.B., Ilxomovich, E.K. Constructing a model of the process of receiving and storing oilseeds in oil and fat enterprises through the Petri net. International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2019.
9. L. Dobrica, E. Niemela. A Survey on Software Architecture Analysis Methods. IEEE Transactions on Software Engineering, 28(7):638-653, 2002.
10. Mo'minov B., Egamberdiyev E. A Neuro-fuzzy model for predicting successful passing of entrance exams of applicants to higher education //Central Asian Journal of Education and Computer Sciences (CAJECS). – 2023. – т. 2. – №. 2. – с. 6-13.
11. P. O. Bengtsson, N. Lassing, J. Bosch, H. V. Vliet. Architecture-Level Modifiability Analysis. Journal of Systems and Software, 69(1-2):129-147, January 2004.



12. B. Lionberger, C. Zhang. ATAM Assistant: A Semi-Automated Tool for the Architecture Tradeoff Analysis Method. Proc. of Software Engineering and Applications Conference, 2007.
13. <http://www.testingfaqs.org/t-static.html>.
14. В. В. Кулямин. Методы верификации программного обеспечения. Конкурс обзорно-аналитических статей по направлению «Информационно-телекоммуникационные системы», 2008.
15. http://en.wikipedia.org/wiki/Automated_theorem_proving.