

Islomov D. Z.

Qarshi davlat universiteti magistranti, O'zbekiston

ARGO UML DASTURI YORDAMIDA AXBOROT TIZIMINI LOYIHALASHTIRISH USULLARI

Annotatsiya: Mazkur maqolada ARGO UML dasturi yordamida axborot tizimini loyihalashtirish usullari, qo'llaniladigan asosiy yondashuvlar, diagrammalar va metodologiyalar, tibbiyot, ta'lim va boshqa sohalarda amaliy qo'llanilishi yoritilgan.

Kalit so'zlar so'zlar: ARGO UML, axborot tizimlari, dasturiy loyihalash, UML diagrammalari, model asosida loyihalash, texnologik integratsiya, sinflar diagrammasi, dasturiy yechimlar, avtomatlashtirilgan tizimlar, loyiha jarayonlari.

Islomov D. Z.

Master's student, Karshi State University, Uzbekistan

INFORMATION SYSTEM DESIGN METHODS USING ARGO UML PROGRAM

Annotation: This article explores the methods of designing information systems using the ARGO UML software, key approaches, applied diagrams and methodologies, as well as their practical applications in fields such as medicine, education, and other domains.

Keywords: ARGO UML, information systems, software design, UML diagrams, model-based design, technological integration, class diagrams, software solutions, automated systems, project processes.

Аннотация: В данной статье рассматриваются методы проектирования информационных систем с использованием программы ARGO UML, основные подходы, применяемые диаграммы и методологии, а также их практическое применение в таких сферах, как медицина, образование и другие области.

Ключевые слова: ARGO UML, информационные системы, программное проектирование, диаграммы UML, проектирование на основе моделей, технологическая интеграция, диаграмма классов, программные решения, автоматизированные системы, проектные процессы.

Kirish:

Bugungi kunda axborot tizimlari nafaqat texnologik inqiloblarni boshqarayotgan, balki jamiyatning barcha sohalarini ham keng qamrovli ravishda o'zgartirmoqda. ARGO UML dasturi yordamida axborot tizimlarini loyihalashtirish zamonaviy texnologiyalarni amaliyotga joriy etishda muhim o'rin tutadi. Ushbu dastur, UML diagrammalari orqali axborot tizimlarini samarali modellashtirish va loyihalashda keng imkoniyatlar yaratadi. Axborot tizimlarining loyihalashtirilishi jarayonida ARGO UML ning o'rni, diagrammalarni tuzish, tizimni optimallashtirish va ilg'or metodologiyalarni qo'llashda katta ahamiyat kasb etadi.

Maqolada ARGO UML dasturi yordamida axborot tizimlarini loyihalashtirish usullari va metodologiyalari batafsil yoritiladi. Shuningdek, ushbu dasturdan foydalanishning afzalliklari, imkoniyatlari va amaliy qo'llanilish sohalari, shu jumladan tibbiyot, ta'lim va boshqa sohalardagi samarali qo'llanilishi misollarida ko'rib chiqiladi. Maqola davomida ARGO UML dasturining axborot tizimlarini loyihalashtirish jarayonidagi o'rni, texnologik integratsiya va raqamli transformatsiya jarayonlari haqida to'liq tahlil keltiriladi.

ARGO UML - bu **ochiq manbali dasturiy ta'minot** bo'lib, UML (Unified Modeling Language) diagrammalarini yaratish va tahlil qilish imkonini beradi. Ushbu dastur axborot tizimlarini loyihalash jarayonini avtomatlashtirish va soddalashtirishga xizmat qiladi.

Asosiy imkoniyatlari:

- UML diagrammalarining turli turlarini yaratish: sinflar diagrammasi, foydalanish holatlari diagrammasi, ketma-ketlik diagrammasi va boshqalar.
- Loyihalashtirilayotgan tizimning strukturasi va funksiyalarini vizual ko‘rinishda taqdim etish.
- Loyihalashtirish jarayonida kutilayotgan xatoliklarni oldindan aniqlash.

ARGO UML yordamida modellashtirishning asosiy afzalliklari

- **Tizimni tahlil qilish imkoniyati:** Diagrammalar orqali tizimning kuchli va zaif tomonlarini aniqlash mumkin.
- **Qulay interfeys:** Foydalanuvchi uchun tushunarli va oson boshqariladigan muhit.
- **Resurslarni tejash:** Loyihalashtirish jarayonida vaqt va mablag‘ni tejashga yordam beradi.
- **Xatoliklarni kamaytirish:** Modellashtirish jarayonida dasturiy ta'minotdagi muammolarni oldindan ko‘rish.

Axborot tizimlarini modellashtirishda ARGO UML yordamida quyidagi diagrammalar keng qo‘llaniladi:

- **Sinflar diagrammasi (Class Diagram):** Tizimning asosiy obyektlarini va ularning o‘zaro bog‘lanishlarini aniqlash.
- **Foydalanish holatlari diagrammasi (Use Case Diagram):** Foydalanuvchilarning tizim bilan o‘zaro aloqasini modellashtirish.
- **Ketma-ketlik diagrammasi (Sequence Diagram):** Operatsiyalar ketma-ketligini aniqlash va ular o‘rtasidagi bog‘lanishni ko‘rsatish.

1. Tizimni tahlil qilish va talablarni aniqlash

Mazmuni:

- Tizimdan kutilayotgan natijalarni belgilash.
- Foydalanuvchilar ehtiyojlari va talablarini o‘rganish.

Misol:

Bir kasalxona bemorlarni boshqarish tizimini yaratmoqchi. Ushbu bosqichda shifokorlar, bemorlar va administratorlarning ehtiyojlari aniqlanadi:

- Shifokorlar uchun tashxis va davolash jarayonlarini kuzatish.
- Bemorlar uchun onlayn ro'yxatdan o'tish va tibbiy ma'lumotlarga kirish.
- Administratorlar uchun xodimlar jadvalini boshqarish.

2. Tizim arxitekturasini loyihalash

- Tizimning asosiy komponentlarini aniqlash.
- Komponentlar o'rtasidagi aloqalarni belgilash.

Misol:

Kasalxona tizimi quyidagi komponentlardan iborat bo'lishi mumkin:

- Foydalanuvchilar (shifokorlar, bemorlar, administratorlar).
- Ma'lumotlar bazasi (bemor ma'lumotlari, tashxislar).
- Tizim interfeysi (veb-sayt yoki mobil ilova).

Diagramma: **Sinflar diagrammasi (Class Diagram)** orqali komponentlar va ularning o'zaro bog'lanishi ko'rsatiladi.

3. Modellashtirish

- Tizimning funktsionalligi va jarayonlari UML diagrammalarida tasvirlanadi.
- Modellashtirish yordamida tizimning ishlash jarayonlari tushunarli bo'ladi.

Misol:

• Foydalanish holatlari diagrammasi (Use Case Diagram):

- Shifokorlar bemorning tibbiy tarixini ko'rish.
- Bemorlar davolanish vaqtlarini tekshirish.
- Administratorlar ro'yxatlarni boshqarish.

• Jarayonlar diagrammasi (Activity Diagram):

- Bemor ro'yxatdan o'tadi.
- Shifokor tashxis qo'yadi.
- Ma'lumotlar bazaga yoziladi.

4. Tizimni amalga oshirish va sinovdan o'tkazish

- Loyihalashtirilgan tizimni dasturlash tillari yordamida ishlab chiqish.
- Sinov jarayonida tizimning ishlashida xatoliklarni aniqlash va ularni tuzatish.

Misol:

- Dasturchilar sinflar diagrammasidan foydalanib tizim kodini yozadilar.
- Har bir jarayon (masalan, foydalanuvchi ma'lumotlarini kiritish) test qilinadi va xatolar aniqlanadi.

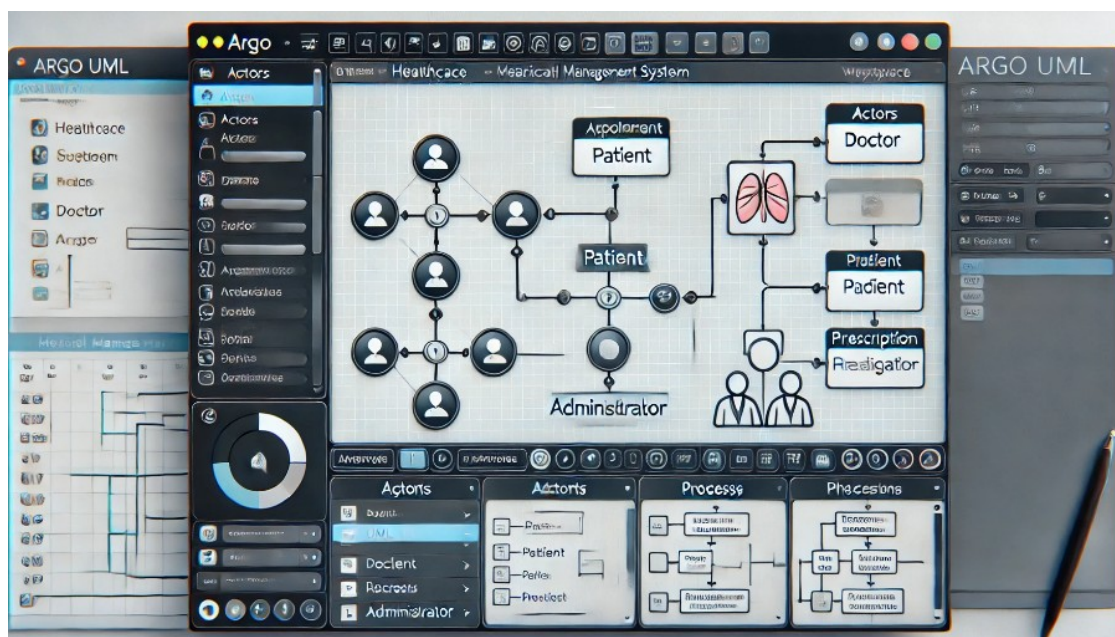
5. Tizimni joriy qilish va qo'llab-quvvatlash

- Tayyor tizimni foydalanuvchilar uchun ishga tushirish.
- Foydalanuvchilardan fikr-mulohazalarni qabul qilish va tizimni takomillashtirish.

Misol:

Kasalxona tizimi ishga tushirilgach:

- Bemorlar va shifokorlar tizimdan foydalanishni boshlaydi.
- Texnik qo'llab-quvvatlash xizmati tizimda yuzaga kelgan muammolarni hal qiladi.



Rasm ARGO UML dasturining ishchi oynasini tasvirlaydi. Ushbu dastur UML diagrammalarini loyihalashtirish uchun mo'ljallangan qulay vositalardan biridir.

1. Ishchi oyna:

- Dastur interfeysining asosiy qismi diagrammalarni tahrirlash uchun mo'ljallangan. Diagramma elementlari markaziy joyda ko'rsatilgan.

- Oynada **Doktor**, **Bemor**, va **Administrator** kabi asosiy aktyorlar aks etgan.

2. **Diagramma:**

- Diagramma tibbiyot sohasidagi axborot tizimini loyihalashtirish jarayonini aks ettiradi.

- Asosiy jarayonlar: **Tibbiy yozuvlarni boshqarish**, **Vaqlarni rejalashtirish**, va **Dori-darmon buyurish** kabi funksiyalarni o'z ichiga oladi.

3. **Qo'shimcha elementlar:**

- Yuqori qismda menyular va vositalar paneli joylashgan, bu diagramma elementlarini tezda qo'shish, tahrirlash yoki o'chirishga yordam beradi.

- Chap tomonda elementlar kutubxonasi, u orqali diagramma komponentlarini loyihaga qo'shish mumkin.

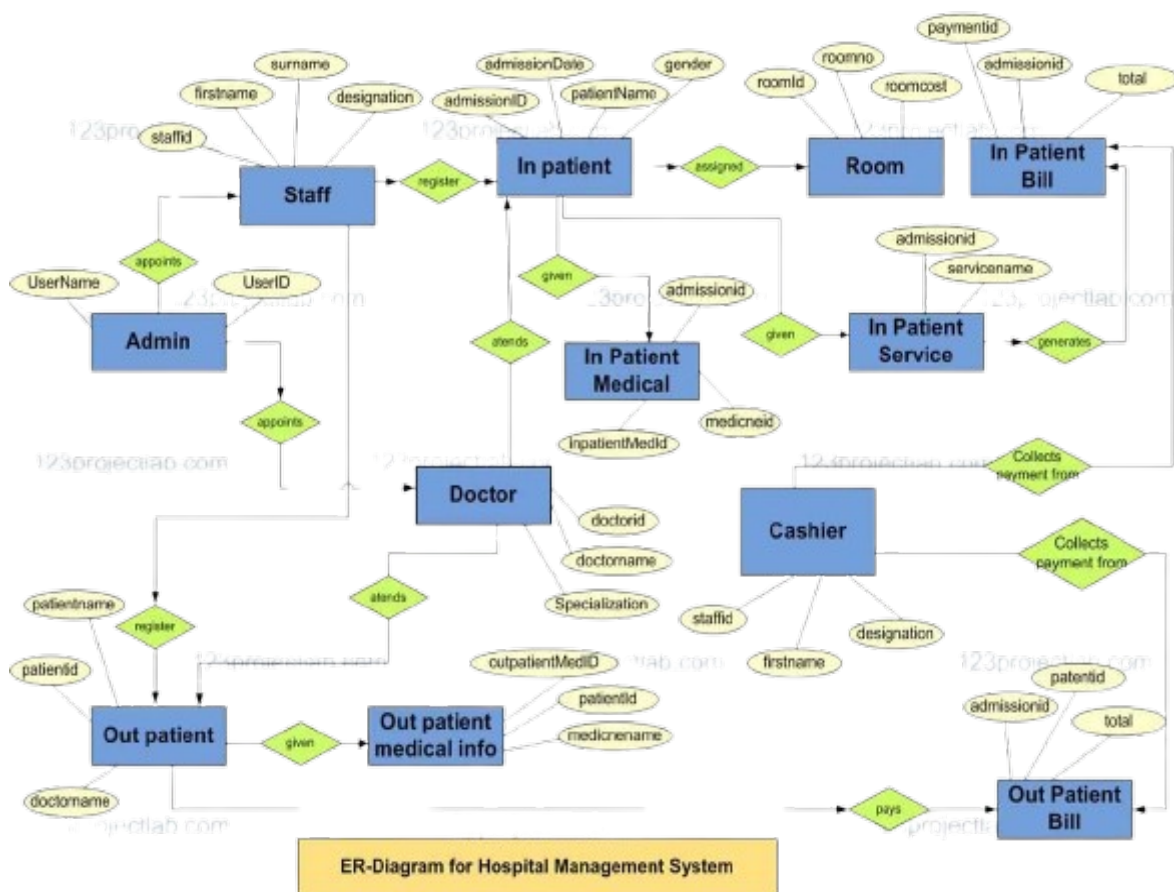


Diagramma Tavsifi:

Loyiha: Tibbiyot muassasasi boshqaruv tizimi

Diagramma turi: UML Use Case Diagram (Foydalanish holatlari diagrammasi)

1. **Personajlar:**
 - o **Bemor (Patient):** Tibbiy xizmatlardan foydalanadi.
 - o **Shifokor (Doctor):** Davolash jarayonlarini boshqaradi.
 - o **Administrator:** Tizimni boshqaradi, bemorlar va shifokorlar jadvalini yuritadi.
2. **Foydalanish holatlari (Use Cases):**
 - o **Ro‘yxatdan o‘tish (Registration):** Bemorlar tizimga kiritiladi.
 - o **Vaqlarni rejalashtirish (Appointment Scheduling):** Shifokor bilan uchrashuvlarni tashkil etadi.
 - o **Tibbiy yozuvlarni boshqarish (Manage Medical Records):** Bemor tarixini saqlaydi.

- **Davolash rejalarini belgilash (Plan Treatment):** Shifokor tomonidan belgilanadi.

- **To'lovlarni boshqarish (Payment Management):** Tibbiy xizmat uchun to'lovlarni amalga oshiradi.

3. **O'zaro bog'liqliklar:**

- Administrator barcha jarayonlarni nazorat qiladi.
- Bemor faqat ro'yxatdan o'tish, uchrashuvlar va to'lovlarni amalga oshiradi.

- Shifokor faqat davolash rejasi va bemor yozuvlari bilan ishlaydi.

Xulosa va takliflar

Maqolada ARGO UML dasturi yordamida axborot tizimlarini loyihalashtirish usullari yoritildi. ARGO UML, tizimlarni vizual tarzda modellashtirishda samarali vosita bo'lib, UML diagrammalarini yaratish orqali loyihalash jarayonini soddalashtiradi. Ushbu dastur yordamida, tibbiyot, ta'lim va boshqa sohalarda tizimlarni loyihalashtirishda samaradorlikni oshirish va mehnat unumdorligini yaxshilash mumkin. Maqolada, ARGO UML dasturining imkoniyatlari va afzalliklari tahlil qilinib, tizimni avtomatlashtirish va optimallashtirish jarayonlari ko'rib chiqildi. Shuningdek, ARGO UML diagrammalari, jumladan, foydalanish holatlari, sinflar va faoliyat diagrammalari yordamida tizimning tuzilishi va jarayonlari aniq ifodalandi. Maqola axborot tizimlarini loyihalashtirishda ARGO UML dasturining samarali qo'llanilishi va uning imkoniyatlarini kengaytirish zaruriyatini ta'kidlaydi.

Adabiyotlar

1. Kurbanov, K. X. (2019). Axborot tizimlari: Nazariy asoslar va amaliyot. Toshkent: "Fan va texnologiya".

2. Saidov, A. N. (2017). Axborot tizimlarining loyihalash metodologiyalari. Toshkent: "Iqtisodiyot va texnologiyalar" nashriyoti.

3. Sharipov, A. S. (2020). Axborot texnologiyalarining rivojlanish tendensiyalari va ularni joriy etish. Toshkent: "O'zbekiston milliy universiteti nashriyoti".

4. Xojiyev, N. O. (2021). Axborot tizimlari va ularning loyihalash usullari. Samarqand: "SamDU".
5. Suleymanov, A. Z. (2018). Dasturiy ta'minot loyihalashda UML texnologiyalari. Toshkent: "Nexus" nashriyoti.
6. Akehurst, D. H., & Norris, M. R. (2006). Modelling Software with UML: An Introduction. Springer.
7. Graham, A., & Soni, A. (2007). UML 2.0 in Practice. Wiley.
8. Egenrieder, D. M., & Bauer, G. (2010). Building Software Architectures with UML 2.x. Springer.
9. Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd Edition, Addison-Wesley.