

УНИВЕРСАЛЬНАЯ ПЛАТФОРМА ДЛЯ ПРОЕКТИРОВАНИЯ GUI И УПРАВЛЕНИЯ ДОСТУПОМ К СПЕЦИАЛИЗИРОВАННЫМ ИНФОРМАЦИОННЫМ СИСТЕМАМ (DSCS)

Набишев Дилшод Пардаевич,
ООО "RWC"

Аннотация. В статье представлена метаданные-управляемая платформа DSCS, предназначенная для быстрого проектирования графических интерфейсов (GUI) и автоматизированных рабочих мест (АРМ), используемых в корпоративных системах управления и учёта. Показана архитектура разделения на Control Plane и User Plane, а также предложена универсальная метамодель, обеспечивающая динамическую генерацию форм, централизованное управление доступом и ведение журналов действий. Практическая ценность платформы заключается в сокращении сроков разработки GUI в 3–5 раз и унификации интерфейсов в корпоративных системах. Схема интеграции DSCS в корпоративных системах показана на Рис.1.

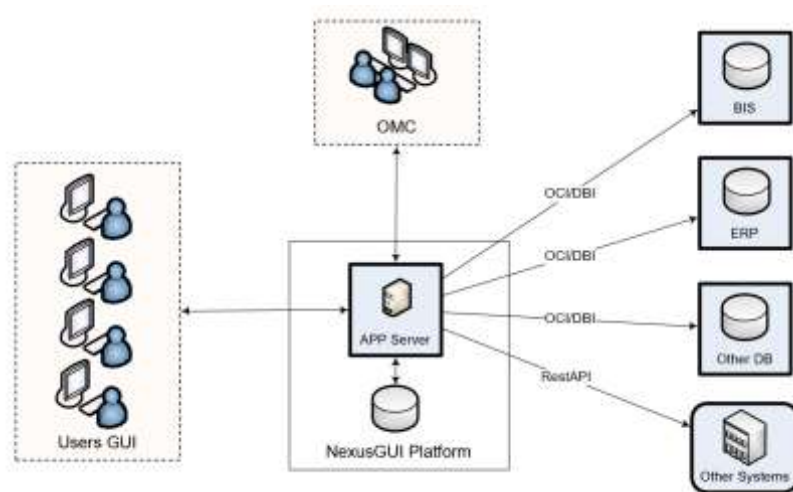


Рис 1.

1. Введение

Большинство корпоративных информационных систем (биллинговых, учётных, ERP, CRM) используют однотипные функции пользовательского интерфейса: ведение справочников, управление пользователями, ролями, аудит действий, загрузку и модификацию данных. Каждый проект вновь реализует эти компоненты, тратя значительные ресурсы.

Платформа **DSCS** позволяет решить эту проблему за счёт вынесения структуры интерфейсов, метаданных, схем доступа, маппинга данных и

аудита в единое хранилище метаданных. GUI-клиент динамически строится на основе описания, а реальные данные загружаются из внешних систем (User Plane) Рис.2. Такой подход обеспечивает повторное использование архитектуры и сокращает трудозатраты разработки.

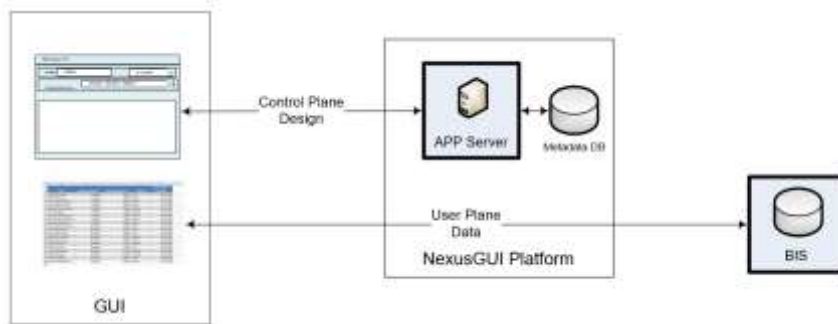


Рис 2.

2. Архитектура DSCS

Архитектура платформы логически разделена на два уровня:

Control Plane — управление интерфейсом

Содержит:

- описания форм (FormDescriptor);
- структуру полей (FieldDescriptor);
- структуры источников данных;
- правила доступа и роли;
- журналирование и аудит;
- административные настройки.

User Plane — доступ к данным прикладной системы

Содержит:

- внешние БД (например, BIS, ERP, CRM);
- API-интерфейсы для выборки и модификации данных;
- сервисы приложений.

GUI-клиент получает структуру интерфейса из Control Plane, а данные — из User Plane. Это позволяет динамически изменять форму без перекомпиляции и переписывания кода.

3. Мета модель данных (общий подход)

Платформа использует внутреннюю метамодель, включающую следующие сущности:

- FormDescriptor — регистрация форм
- FieldDescriptor — структура полей и типов
- DataSourceDescriptor — источники данных
- RoleMapping — привязка ролей к формам
- AuditLog — запись действий пользователя

- UserDescriptor — настройка пользователей

Эта модель позволяет динамически генерировать интерфейсы, обеспечивая гибкость и расширяемость.

4. Научно-развёрнутое описание административного модуля

Административный модуль является ключевым компонентом Control Plane и обеспечивает управление пользователями, ролями, аудитом безопасности и структурой организации. Он основан на формальной метамодели, включающей четыре функциональных кластера данных.

4.1. Таблицы управления пользователями

Обеспечивают хранение и структурирование данных о пользователях и их организационной принадлежности:

- ADM_USERS — основные данные пользователей;
- ADM_DEPARTMENTS — структура подразделений;
- ADM_OFFICES — офисы и географические единицы;
- ADM_HOST_USER — соответствие пользователей

оборудованию или хостам доступа.

4.2. Таблицы управления доступом

Отвечают за ролевую модель платформы:

- ADM_ROLES — список ролей системы;
- ADM_USER_ROLES — назначение ролей пользователям;
- ADM_WWW_ARM — список доступных APM (GUI-модулей)

системы.

4.3. Таблицы аудита и безопасности

Формируют полную трассируемость действий пользователей:

- ADM_ACTION_LOG — журнал действий;
- ADM_ACTION_MIC_LOG — детализация транзакций;
- ADM_ACTION_TYPES — классификатор действий;
- ADM_ERROR_LOG — лог ошибок;
- ADM_USER_SESSIONS — учёт сеансов работы.

4.4. Технические таблицы интеграции

Используются для технического контроля и безопасности:

- ADM_HOSTS — перечень хостов входа;
- ADM_HOST_STATUS — состояние хостов.

5. Пример метаданных интерфейса

Фрагмент JSON-описания формы:

```
{
  "formId": 102,
  "title": "Абоненты",
```

```
"fields": [
  {"name": "MSISDN", "type": "string", "label": "Номер"},
  {"name": "Status", "type": "enum", "ref": "RefStatus"}
],
"dataSource": "bis.subscriber",
"permissions": ["READ", "UPDATE"]
}
```

Этот фрагмент демонстрирует метаданные-управляемый GUI, где не требуется код для изменения интерфейса.

6. Новизна исследования

Концептуальная новизна:

- Разделение на Control Plane и User Plane для корпоративных GUI.
- Полностью метаданные-управляемый механизм построения интерфейсов.
- Унифицированная административная подсистема, включающая ролевую модель и аудит.

Технологическая новизна:

- Использование универсальной метамодели Form-Field-DataSource-Role.
- Динамическая генерация GUI без перекомпиляции.
- Централизованное описание структуры всех АРМ.

Практическая значимость:

- Сокращение сроков разработки GUI на 3–5 раз.
- Единообразие интерфейсов в корпоративных системах.
- Уменьшение количества ошибок за счёт унифицированного Control Plane.

7. Заключение

Платформа **DSCS** предлагает универсальный подход к построению интерфейсов для корпоративных систем, основанный на концепции метаданных и двухуровневой архитектуре управления. Решение повышает скорость разработки, обеспечивает гибкость и стандартизацию АРМ, что делает его применимым в крупных информационных системах, биллингах и системах корпоративного учёта.

8. Список литературы

1. Фаулер М. *Архитектура корпоративных программных решений*. – М.: Вильямс, 2020. – 784 с.
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. *Приемы объектно-ориентированного проектирования. Паттерны проектирования*. – М.: Питер, 2019. – 416 с.

3. Буч Г. *Объектно-ориентированный анализ и проектирование*. – М.: Вильямс, 2018. – 560 с.
4. Newman S. *Building Microservices: Designing Fine-Grained Systems*. – O'Reilly Media, 2021. – 360 p.
5. Bernstein P.A., Newcomer J. *Principles of Transaction Processing*. – Morgan Kaufmann, 2020. – 608 p.
6. van der Aalst W.M.P. *Process Mining: Data Science in Action*. – Springer, 2022. – 507 p.
7. ISO/IEC 25010:2011. *Systems and software engineering — Systems and software quality models*.
8. Fielding R.T. *Architectural Styles and the Design of Network-based Software Architectures*. – University of California, 2000. – 162 p.
9. Martin R.C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. – Prentice Hall, 2017. – 432 p.
10. Kosslyn S.M. *Graph Design for the Eye and Mind*. – Oxford University Press, 2006. – 248 p.