



LUI - Эльбрус. Новое средство коллективной разработки приложений для PostgreSQL

Для создания сложных интерактивных WEB-приложений теперь достаточно знания языка SQL

Группа компаний «ФОРС» представляет современный программный продукт – интероперабельное инструментальное средство для коллективной разработки прикладных систем – Live Universal Interface (LUI).

По результатам успешных испытаний LUI на серверах Эльбрус зарегистрирован программный продукт **LUI-Эльбрус** в Федеральной службе по интеллектуальной собственности и Едином реестре российских программ для ЭВМ и БД (№7692).

Для систем, основанных на PostgreSQL, LUI выполняет роль подобную APEX и Forms. Концепции и технологии, применяемые в LUI, делают этот инструмент носителем уникальных качеств – как для конечного пользователя, так и для разработчиков приложений.

Средства разработки сегодня

Несмотря на огромный выбор инструментальных средств разработки ПО, часто возникает ситуация, когда создание пользовательского интерфейса требует больше ресурсов, чем проектирование и реализация прикладной бизнес-логики. В результате проект может начать «теряться» в экранных формах.

Программисты вынуждены самостоятельно строить алгоритмы для обработки событий при вводе и отображении данных и управлять распределением информации на экране, неся полную ответственность за результат. Это требует от проектной команды обладания широким набором компетенций, что повышает стоимость разработки и снижает рентабельность проектов.

В чем новизна решения

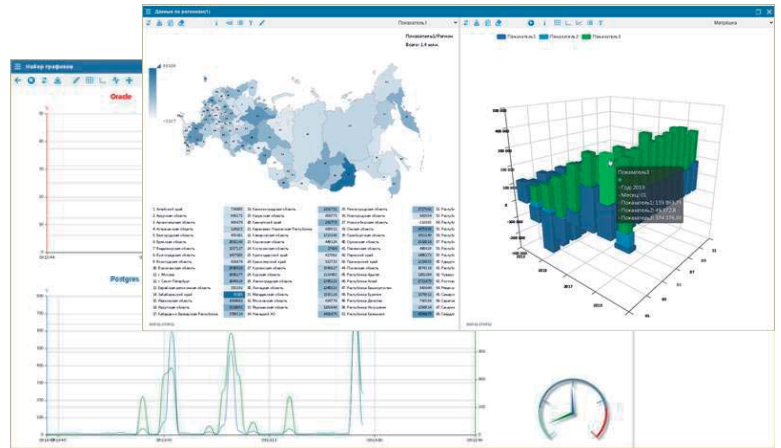
LUI предлагает совершенно иной подход, при котором разработка прикладной логики системы отделена от разработки пользовательского интерфейса.

Обе эти задачи могут решаться независимо, а при создании экранных форм достаточно правильно декларировать определённые свойства элементам интерфейса.

В результате все необходимые операции будут выполняться автоматически, не требуя программной обработки событий.

Возможности LUI

- Быстрое и простое создание экранных интерфейсов приложений, в том числе, для давно работающих систем.
- Реализация в кратчайшие сроки учётных, финансовых и расчётных систем для любых отраслей.
- Обеспечение разграничения прав доступа пользователей, аудит и администрирование их действий, мониторинг состояния СУБД.
- Коллективная разработка: DevOps, встроенная система управления версиями, мощные средства отладки.
- Поддержка реальной многоязычности. Перевод существующего приложения LUI на многоязычность одной командой.
- Наличие встроенных средств автоматизированного создания интерфейсных форм для доступа к уже имеющимся сущностям прикладной системы.
- Группировка данных с применением функций над критериями агрегирования и агрегируемыми данными. Автоматическое представление результатов в интерактивных диаграммах, включая трёхмерные диаграммы и возможности детализации ("drill-down").
- Query by Example (QBE) с учётом специфических типов данных, с настройкой на пользовательские типы данных без программирования.
- Реальный гетерогенный мультиконнект как в одну базу данных, так и в различные базы, в том числе, разных производителей. Возможность разработки приложения с «короткими» и «длинными» транзакциями.
- Настройка приложения на специфические типы данных, использование модулей-корректоров и «раскраска» текста при вводе, «на лету».



Преимущества

- Существенное снижение затрат на разработку экранных интерфейсов приложений (в 3 и более раз).
- Прямое внесение в работающую систему любых изменений в экранном интерфейсе без её остановки: этапы подготовки локальных модификаций, верификации изменений, компиляции и сборки не требуются.
- Создание дополнительных потребительских качеств у приложения - например, сочетание традиционных свойств «тонкого» клиента с поддержкой «длинных» транзакций.
- Увеличение продолжительности жизни приложения, поскольку среда реализации пользовательского интерфейса может меняться, не затрагивая логику системы.
- Удобное облачное средство разработки сложных облачных приложений.
- Реальное масштабирование по трём направлениям: база данных, сервер приложения и сервер отображения.

Особенности

- Использование декларативных языков при описании свойств элементов интерфейса.
- Абстрагирование свойств элементов интерфейса от средств их реализации.
- Переход от событийной к декларативной разработке программной логики.
- Оперативное подключение к любому существующему и работающему приложению.