



ЛОЦМАН: PLM Архитектурное проектирование

Описание функциональных характеристик экземпляра программного обеспечения с описанием процессов, микроконтейнеров, расположения файлов ПО, его компонентов (в порядке пункта 11 «а» Правил формирования и ведения единого реестра российского ПО и единого реестра ПО из государств-членов ЕАЭС, за исключением РФ, утв. постановлением Правительства РФ от «16» ноября 2015 г. № 1236)

ООО «ПЛМ УРАЛ»

Россия, г. Екатеринбург

2024 год

ЛОЦМАН:PLM Архитектурное проектирование

Описание функциональных характеристик

г. Екатеринбург

2025 г.

Содержание

Содержание.....	2
Принятые сокращения и предметные термины	3
1 Введение	4
2 Предоставляемые функциональные возможности	5
2.1 Разработка архитектурных моделей сложных технических систем	5
2.1.1 <i>Методологическое руководство</i>	6
2.1.2 <i>Семантическая цветовая карта</i>	7
2.1.3 <i>Поддержка онтологии метода Arcadia</i>	8
2.1.4 <i>Семантический браузер.....</i>	8
2.1.5 <i>Семантическое удаление с предварительным просмотром</i>	9
2.1.6 <i>Проверка модели.....</i>	9
2.1.1 <i>Вычисляемое отношение</i>	10
2.1.2 <i>Расширенное управление диаграммами.....</i>	11
2.1.3 <i>Поддерживаемые типы диаграмм</i>	11
2.2 Интеграция архитектурных моделей в общую PLM-экосистему предприятия	15
3 Правила установки.....	20
4 Правила эксплуатации.....	21
5 Программные компоненты и совместимость	22
6 Технологические стеки (компоненты).....	23

Принятые сокращения и предметные термины

Термин/сокращение	Расшифровка
ПО	Программное обеспечение
EPL	Eclipse Public License
OSI	Open Source Initiative (www.opensource.org)
OSS	OpenSourceSoftware
Apache License	Лицензия на свободное программное обеспечение Apache Software Foundation.
Eclipse Public License	Лицензия открытого программного обеспечения, которая используется Eclipse Foundation для своих продуктов.
Arcadia	(Architecture Analysis and Design Integrated Approach) – представляет собой метод разработки системной и программной архитектуры, основанный на модели-ориентированной системной инженерии.

1 Введение

Программное обеспечение ЛОЦМАН:PLM Архитектурное проектирование для ЛОЦМАН:PLM (далее ЛОЦМАН:PLM АП, ПО) – это первая в своем роде отечественная интегрированная инженерная среда, обеспечивающая цифровую связь между процессами системной инженерии и классического проектирования (разработка РКД), которую инженеры могут использовать для реализации подходов модели-ориентированной системной инженерии к процессам проектирования.

2 Представляемые функциональные возможности

ЛОЦМАН:PLM Архитектурное проектирование помогает инженерам четко и формализовано описать принципы реализации системы, ее компонентов, а также их взаимодействие как между собой, так и с окружающей средой. Эти описания формируются в виде формальных моделей, которые строго соответствуют методу Arcadia.

ЛОЦМАН:PLM Архитектурное проектирование позволяет проектировщикам связывать процессы управления требованиями и разработки продукта на основе моделей с остальной инженерной информацией. Это в свою очередь и отличает ЛОЦМАН:PLM Архитектурное проектирование от существующих автономных инструментов моделирования на языках UML/SysML, которые создают архитектуру системы/изделия отдельную от всей последующей инженерной информации.

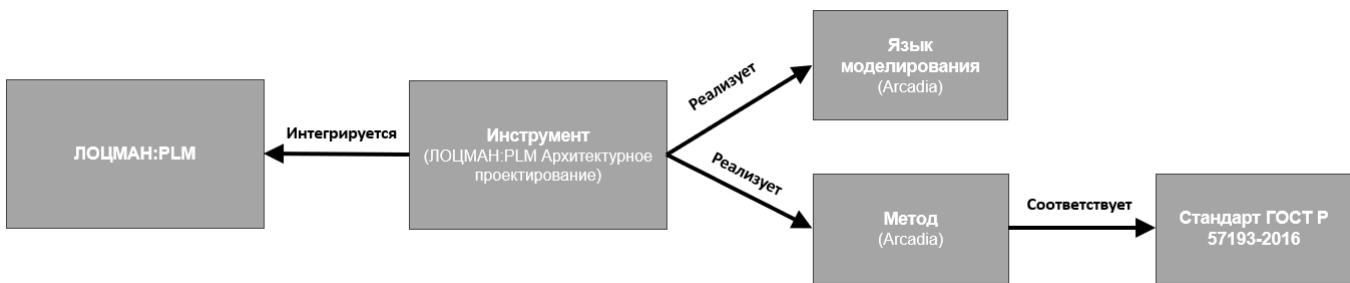


Рисунок 1—Схема, показывающая реализацию языка и метода Arcadia в ПО ЛОЦМАН:PLM Архитектурное проектирование, а также интеграцию с ЛОЦМАН:PLM

Основным функциональным назначением ЛОЦМАН:PLM Архитектурное проектирование является:

1. Разработка архитектурных моделей сложных технических систем (изделий).
2. Управление и документирование знаниями.
3. Интеграция архитектурных моделей в общую PLM-экосистему предприятия.

2.1 Разработка архитектурных моделей сложных технических систем

Основываясь на проверенных на практике и открытых методологических рекомендациях и инструментах, ЛОЦМАН:PLM Архитектурное проектирование помогает проектировщикам технических систем лучше:

- Понимать потребности клиентов.

- Определять и делиться решениями между заинтересованными сторонами.
- Обеспечивать инженерное сотрудничество.
- Выполнять оценку системной архитектуры на ранних этапах проектирования, чтобы сделать правильный выбор из всего множества возможных вариантов.
- Выполнять верификацию и валидацию проектируемой системы.

Все вышеописанное достигается за счет следующего функционала:

- Русифицированный и интуитивно понятный пользовательский интерфейс, позволяющее инженерам графически представлять сложные продукты и связанные с ними системы.
- Постоянная методологическая поддержка пользователя, за счет встроенно-методологического руководства.
- Наличие семантической цветовой карты для объектов разного типа.
- Прослеживаемость между объектами в горизонтальном и вертикальном направлении, за счет поддержки онтологии метода Arcadia.
- Полный анализ влияния в нескольких областях для быстрого поиска инженерных артефактов и навигации по ним за счет применения семантического браузера функционала семантического удаления с предварительным просмотром.
- Проверка разработанной модели на согласованность и соответствие методу Arcadia.
- Представление информации о моделях в нужном для конкретной заинтересованной стороны виде, за счет инструментов упрощения и фильтрации диаграмм.
- Создание системных моделей для представления различных функциональных, логических и физических потребностей, за счет наличия набора поддерживаемых типов диаграмм.

2.1.1 Методологическое руководство

Обозреватель действий обеспечивает методический доступ ко всем ключевым действиям ЛОЦМАН:PLM Архитектурное проектирование. Это главная точка входа в модель, которая предназначена как для новичков, так и для опытных пользователей.

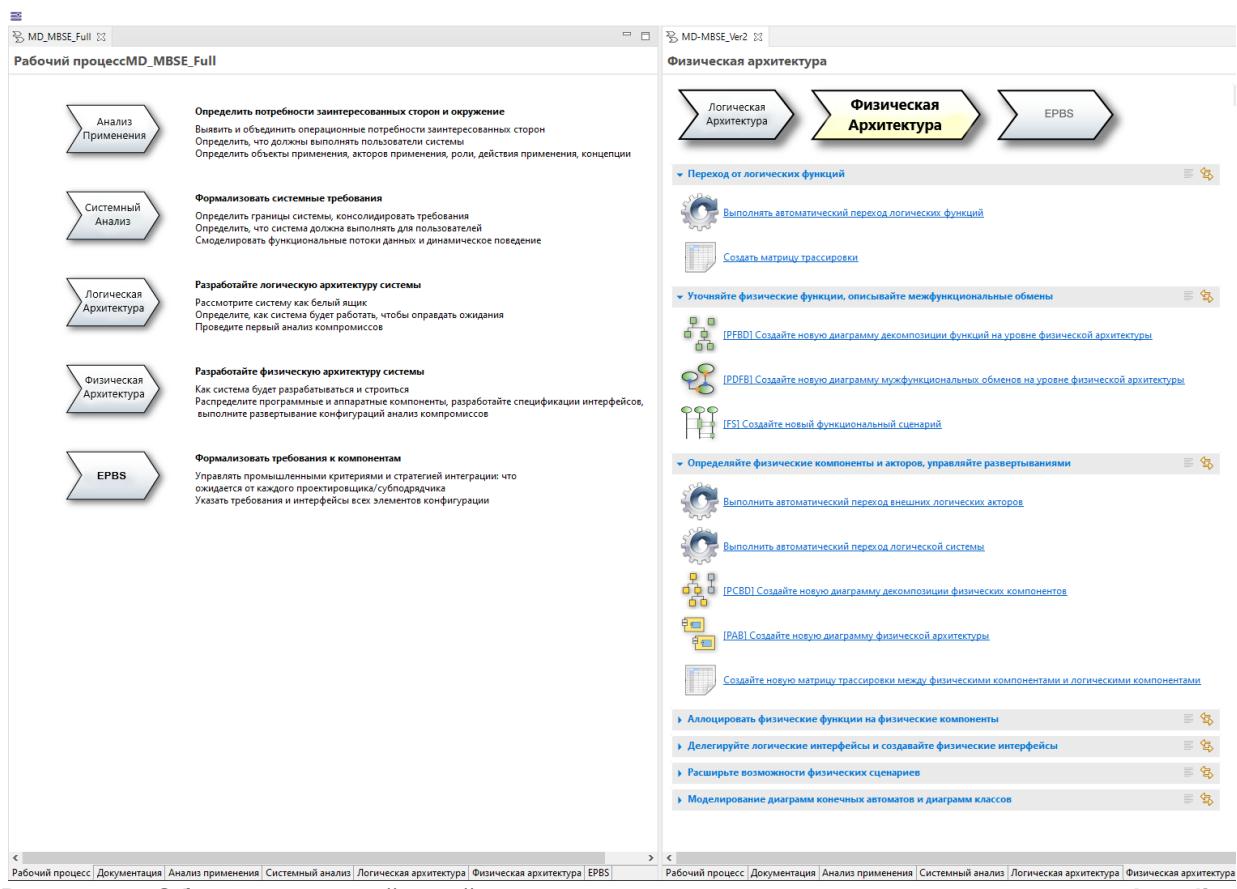


Рисунок 2– Обозреватель действий с методическим руководством по методологии Arcadia

2.1.2 Семантическая цветовая карта

Поскольку графические представления элементов играют ключевую роль в коммуникации, ЛОЦМАН:PLM АП опирается на согласованную цветовую схему. Это способствует повышению читаемости модели для всех заинтересованных сторон (архитекторов, специалистов по верификации и валидации, инженеров по специальностям, менеджеров и т.д.).



Рисунок 3– Семантическая цветовая карта

2.1.3 Поддержка онтологии метода Arcadia

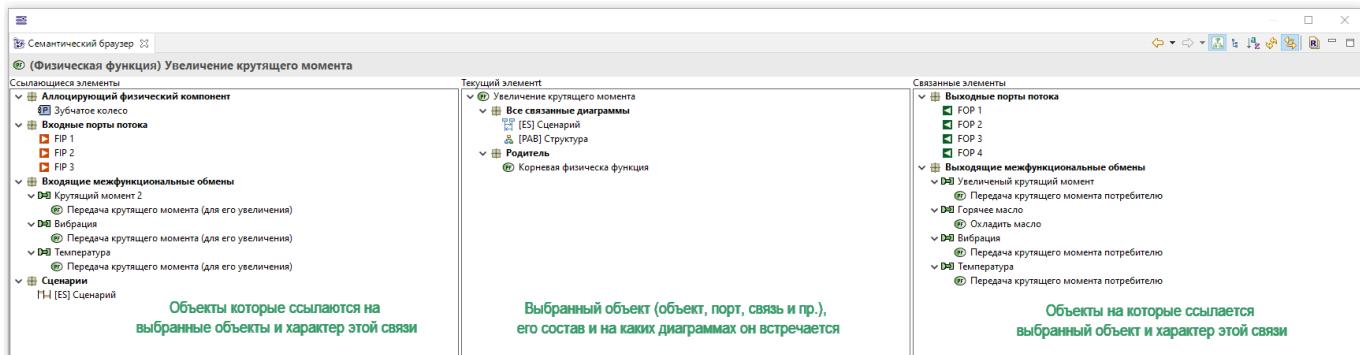
Программное обеспечение ЛОЦМАН:PLM Архитектурное проектирование обеспечивает реализацию онтологии, заложенную в методе Arcadia. Благодаря чему достигается согласованность в определении архитектуры, проектируемой системы/изделия. Каждый слой метода и элементы модели связаны с элементами предыдущего и следующего уровней с помощью связей реализации. Эта функция является ключевой, так как модель может отражать потребности пользователя на верхнем уровне, определять ожидания системы на следующем и затем описывать, как система будет работать и строиться на физическом уровне. Модель предоставляет возможность перемещаться по различным связям реализации и обеспечивать обоснование заложенных в ней идей.

Например, пройдя по связям реализации, можно:

- Понять и проверить, какие функции были получены из более высоких слоев.
- Определить, какие структурные элементы, такие как компоненты системы, выполняют определенные функции.
- Изучить описание возможностей, включая функции, взаимодействия, режимы и состояния. Это помогает разработчику модели лучше понять, какие функции ему нужны и как они будут взаимодействовать.
- Увидеть интерфейсы между различными компонентами и визуально оценить, нет ли каких-либо конструктивных недостатков с точки зрения различных заинтересованных сторон.

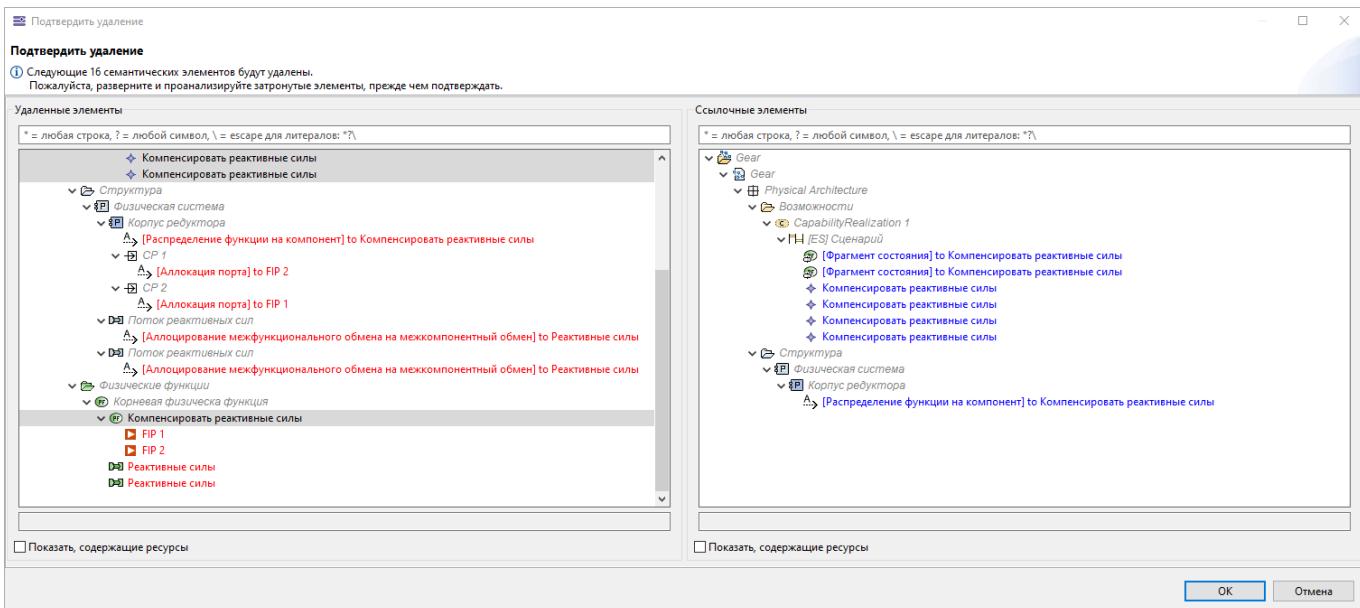
2.1.4 Семантический браузер

Более полезный, чем стандартное иерархическое представление модели, семантический браузер мгновенно предоставляет контекст элементов модели через запрос. Это предпочтительный способ навигации по моделям и диаграммам, а также для проведения быстрого анализа взаимосвязей между элементами модели.


Рисунок 4 – Семантический браузер

2.1.5 Семантическое удаление с предварительным просмотром

ЛОЦМАН:PLM АП обеспечивает мгновенный анализ последствий удаления объектов.


Рисунок 5 – Окно анализа последствий удаления объекта из модели

2.1.6 Проверка модели

ЛОЦМАН:PLM АП организует правила проверки модели в нескольких категориях: проектирование, целостность, качество и т.д. Архитекторы могут определять профили проверки, фокусируясь на различных аспектах. Функционал быстрых исправлений по возможности обеспечивает быстрые и автоматизированные решения.

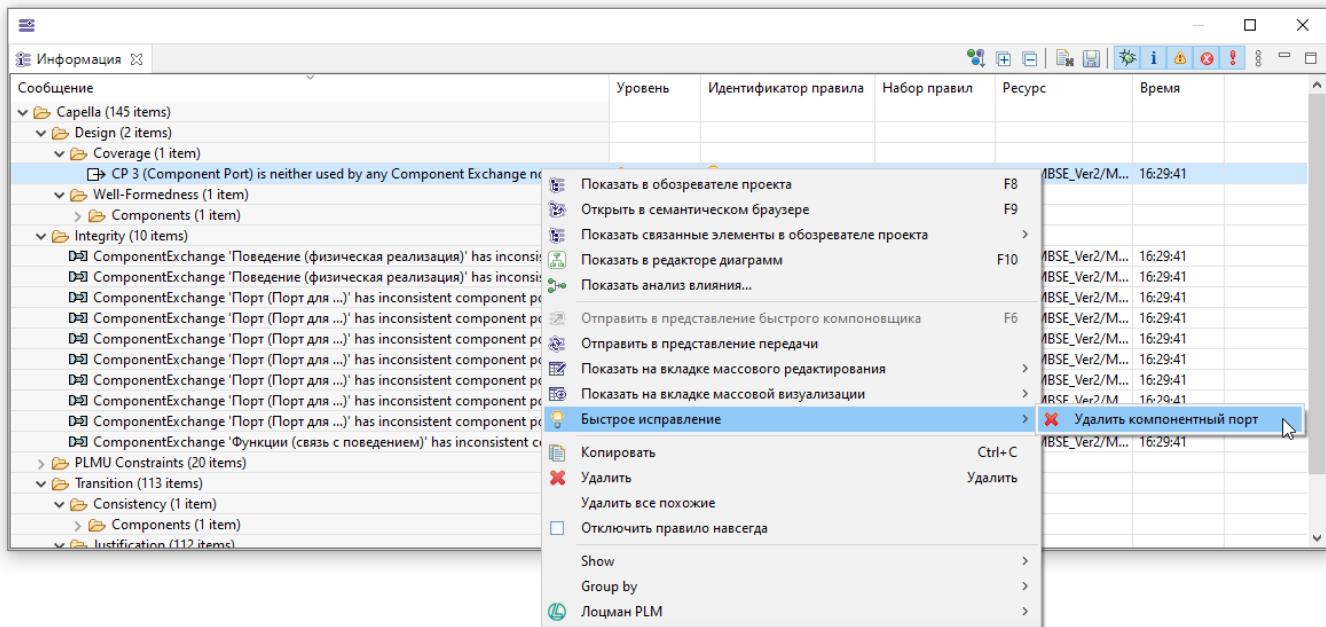


Рисунок 6 – Окно вывода результатов проверки модели

2.1.1 Вычисляемое отношение

Ключевой особенностью приложения ЛОЦМАН:PLM АП является его способность масштабировать и справляться с возрастающей сложностью моделей. ЛОЦМАН:PLM АП автоматически вычисляет графические упрощения. Обмен информацией между функциями более низкого уровня автоматически отображается на функциях более высокого уровня. Данная функциональность освобождает архитекторов от управления промежуточным обменом информацией и ручного поддержания согласованности между несколькими уровнями абстракции. ЛОЦМАН:PLM АП также предоставляет основанные на тегах механизмы, позволяющие графически группировать семантически близкие обмены.

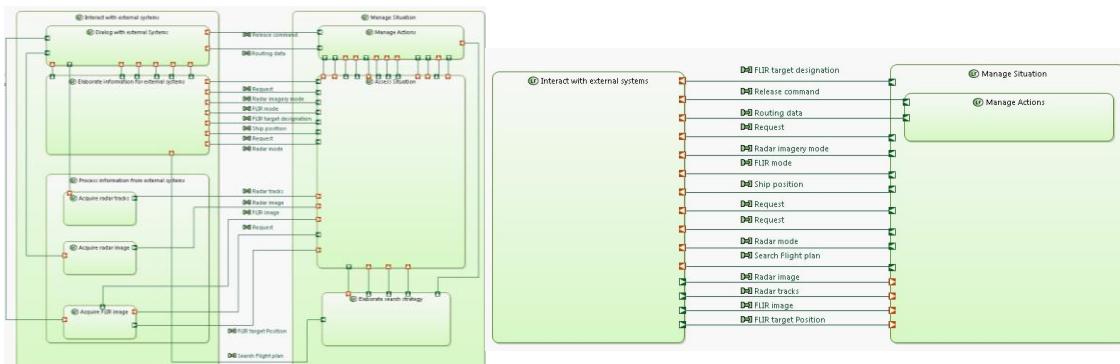


Рисунок 7 – Пример упрощения отображений на диаграммах

2.1.2 Расширенное управление диаграммами

Автоматизированные контекстные диаграммы: содержимое автоматически обновляется в соответствии с предварительно выбранными элементами и предопределенными семантическими правилами.

Синхронизация/десинхронизация диаграмм: обеспечивает тонкий уровень контроля над элементами, которые систематически отображаются или нет (порты функций, межкомпонентные обмены, свойства классов и т.д.). Фильтры помогают улучшить читаемость диаграмм путем выбора параметров отображения и автоматического скрытия/показа элементов. Слои – это средства настройки графического отображения элементов на диаграмме на основе различных интересов (например, специальных точек зрения).

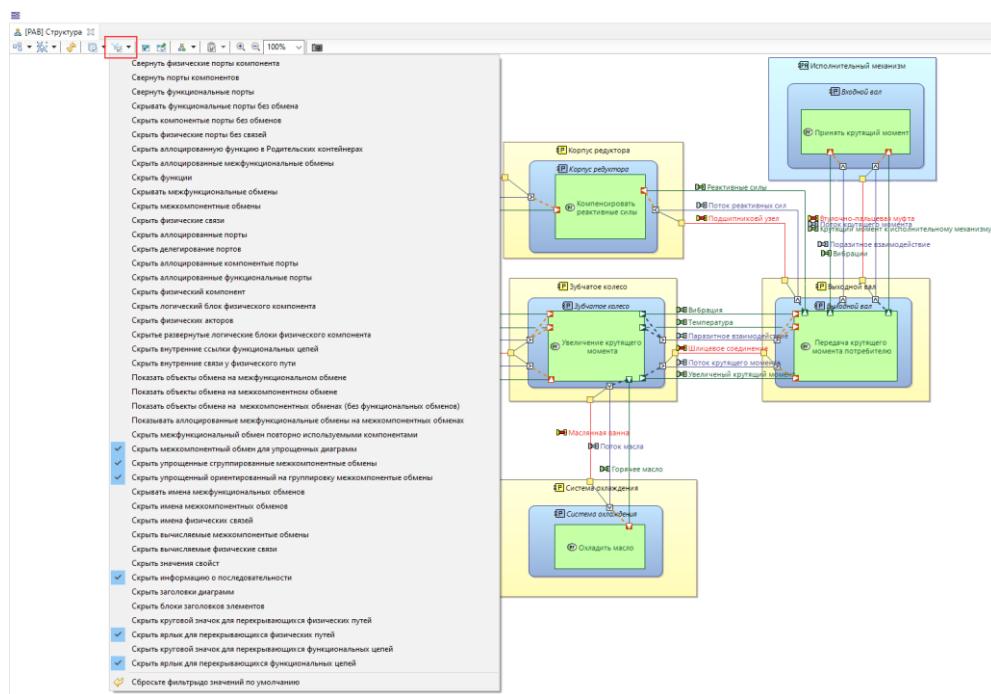


Рисунок 8 – Фильтры диаграммы

2.1.3 Поддерживаемые типы диаграмм

2.1.3.1 Классы и интерфейсы

ЛОЦМАН:PLM АП представляет передовые механизмы для моделирования структур данных с точностью до бита и связывания их с межфункциональными обменами, портами компонентов или функций, интерфейсами и т.д.

2.1.3.2 Режимы и состояния

Диаграммы режимов и состояний – это конечные автоматы, вдохновленные UML. Режимы/состояния/переходы могут быть связаны с функциями, межфункциональными обменами, операциями интерфейсов и т.д.

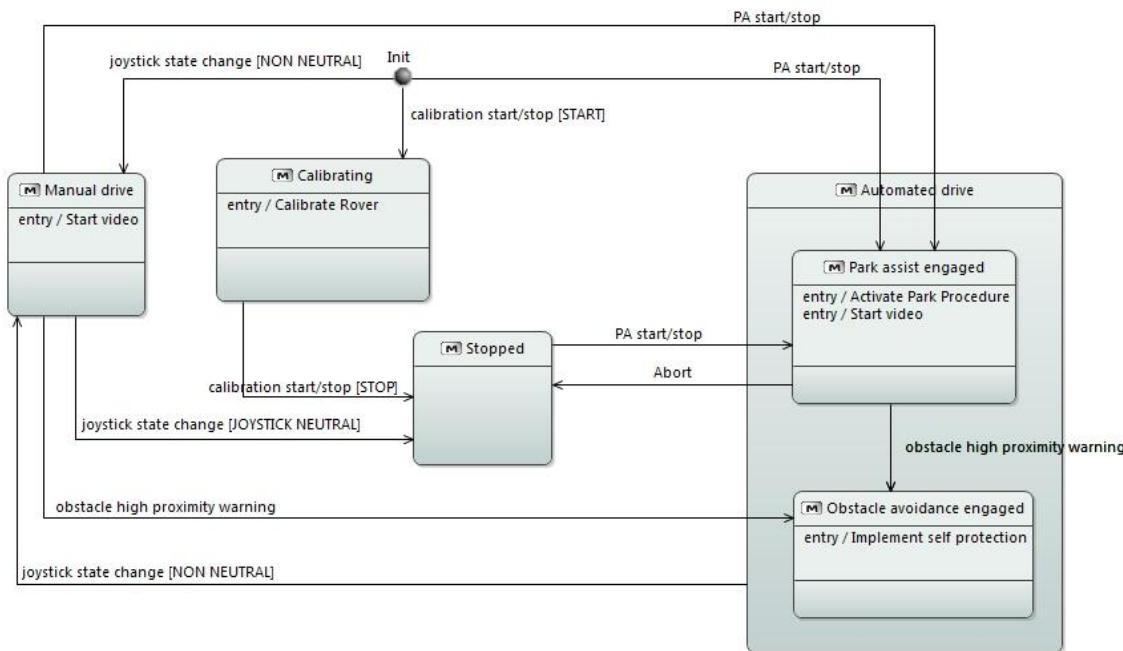


Рисунок 9 - Пример диаграммы режимы и состояния

2.1.3.3 Возможности

Диаграммы возможностей доступны на всех архитектурных уровнях Arcadia, но особенно они полезны на уровнях анализа применения и системного анализа. Возможности аналогичны вариантам использования в UML. Они иллюстрируются потоками данных, функциональными цепями и диаграммами последовательностей.

2.1.3.4 Диаграммы последовательности (Сценарии)

ЛОЦМАН:PLM АП представляет несколько видов диаграмм последовательностей: функциональные сценарии (линии жизни являются функциями), сценарии обмена (линии жизни являются компонентами/акторами, а сообщения последовательности являются межфункциональными или межкомпонентными обменами), сценарии интерфейсов (линии жизни являются компонентами/акторами, сообщения последовательности являются операциями интерфейсов). Режимы, состояния и функции также могут быть отображены на этих диаграммах.

2.1.3.5 Потоки данных

Диаграммы потоков данных доступны на всех архитектурных уровнях Arcadia. Они представляют информационную зависимость между функциями. Эти диаграммы представляют богатые механизмы для управления сложностью: вычисляемые упрощенные отношения между высокоуровневыми функциями, категоризация обменов и т.д. Функциональные цепи могут отображаться на данных диаграммах как выделенные пути.

2.1.3.6 Архитектура

Архитектурные диаграммы используются на всех архитектурных уровнях Arcadia. На уровне анализа применения они фиксируют аллокацию действий применения по сущностям и акторам применения. Процессы применения могут быть отображены в виде выделенных путей.

На остальных архитектурных уровнях их главная цель - показать аллокацию функций по компонентам. Функциональные цепи могут быть отображены как выделенные пути. На уровне системного анализа эти диаграммы содержат один блок, представляющую изучаемую систему, а также блоки представляющие акторов.

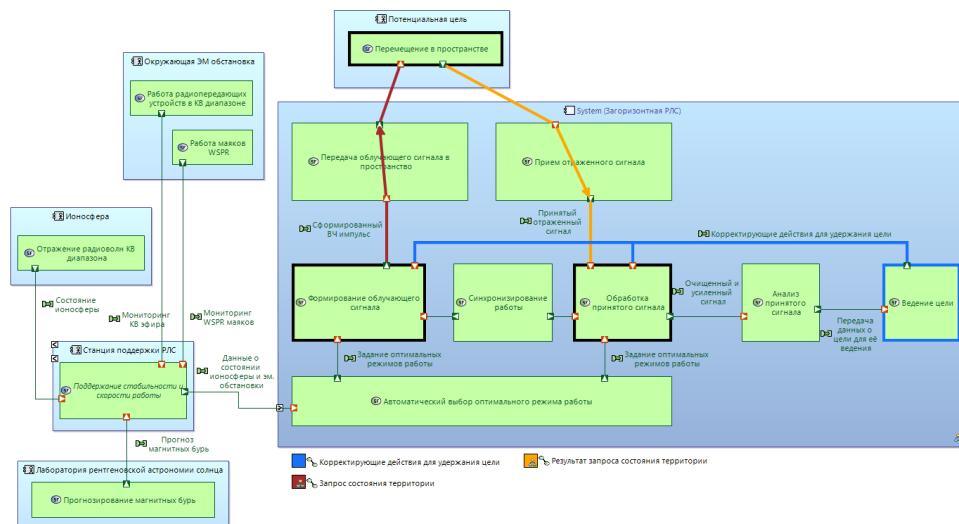


Рисунок 10 - Пример диаграммы системной архитектуры

На уровне логической архитектуры эти диаграммы показывают строительные блоки системы. Они называются логическими компонентами.

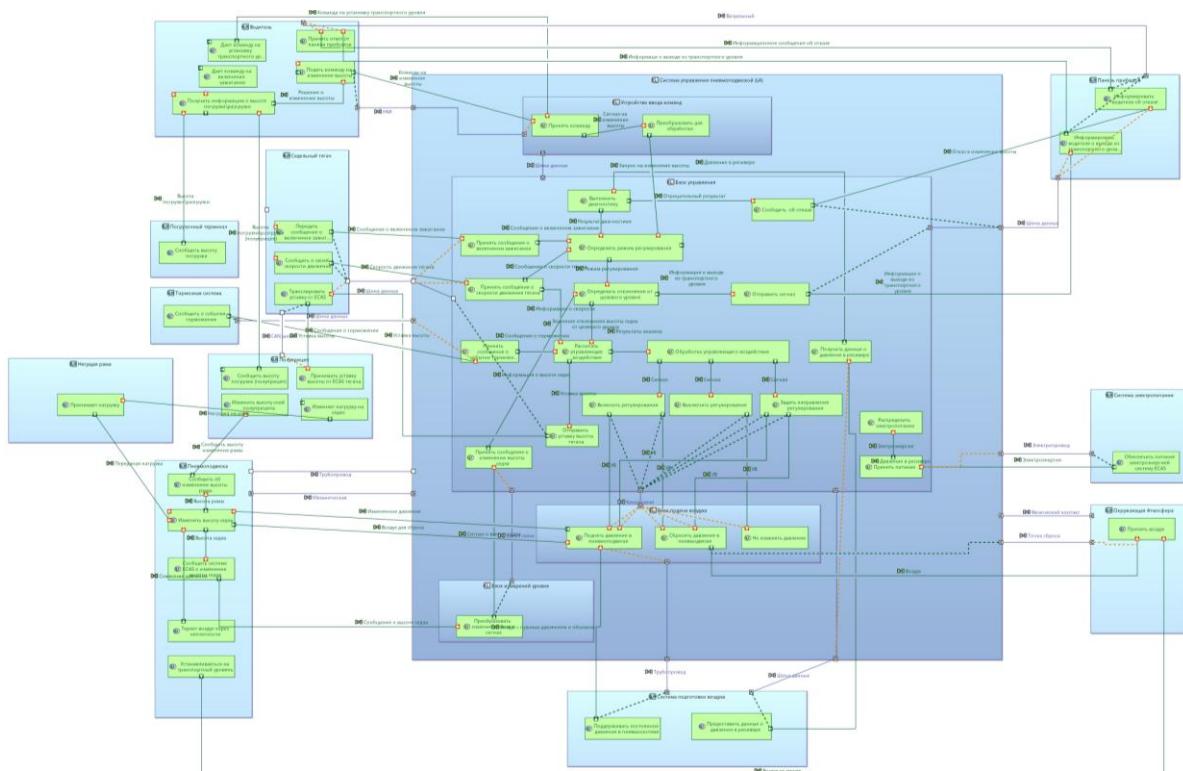


Рисунок 11 - Пример диаграммы логической архитектуры

В физической архитектуре эти диаграммы также показывают распределение компонентов поведения по компонентам реализации (обычно материальным, но не обязательно).

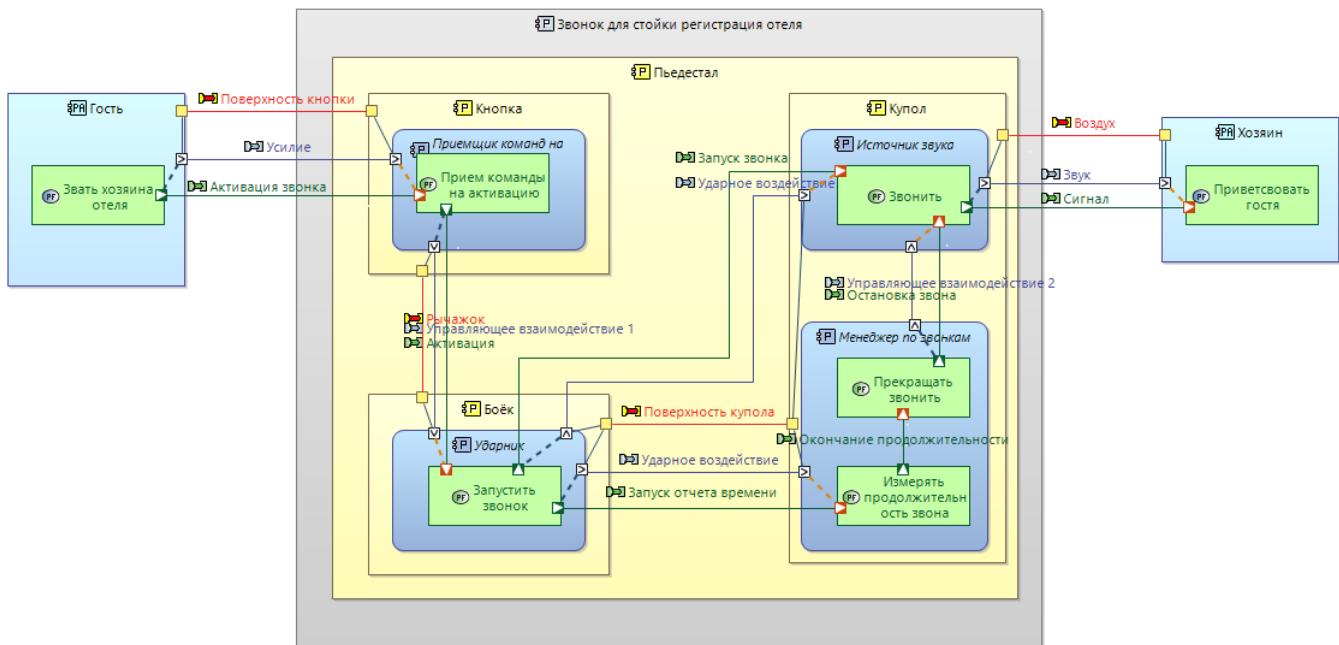


Рисунок 12 - Пример диаграммы физической архитектуры

2.1.3.7 Деревья (Иерархия)

Древовидные или, другими словами, иерархические диаграммы представляют собой разбиение либо функций, либо компонентов. Каждый узел может быть свернут/развернут.

2.2 Интеграция архитектурных моделей в общую PLM-экосистему предприятия

Интеграция ЛОЦМАН:PLMAрхитектурное проектирование с PLM-решением ЛОЦМАН:PLM, позволяет выполнять разработку системной архитектуры изделия согласно методологии Arcadia, которая в дальнейшем будет основой для последующих стадий процесса проектирования в рамках PLM концепции, позволяя создавать комплексного цифрового двойника.

В отличии от конкурирующих инструментов, интегрированный с ЛОЦМАН:PLM, ЛОЦМАН:PLMAрхитектурное проектирование позволяет создавать архитектурные решения и видеть, как они реализуются на последующих стадиях проектирования, тем самым позволяя быстро вносить корректировки в архитектуру с целью разработать действительно надежную техническую систему.

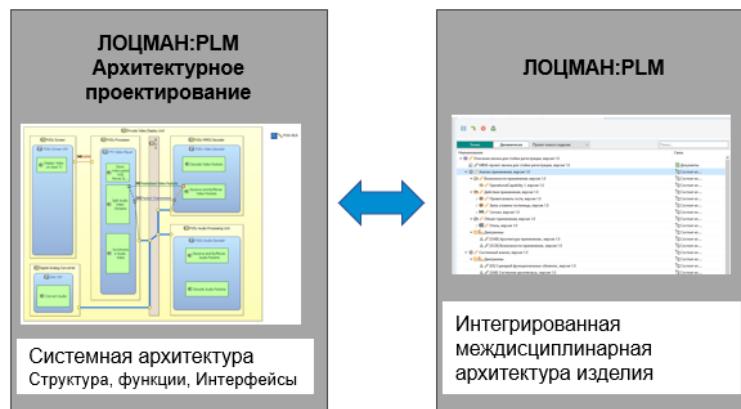


Рисунок 13 – Схема интеграции системной архитектуры в общую PLM-экосистему предприятия

К основные возможностям программного обеспечения ЛОЦМАН:PLM Архитектурное проектирование в части интеграции с ЛОЦМАН:PLM можно отнести:

- Хранение и открытие файла, хранящего описание архитектуры в ЛОЦМАН:PLM (рисунок 14). ;
- Трассировка требований и характеристик между ЛОЦМАН:PLM и ЛОЦМАН:PLM АП (рисунок15);

- Открытие объектов из ЛОЦМАН:PLM АП в ЛОЦМАН:PLM и наоборот (рисунок 16);
- Сохранение в ЛОЦМАН:PLM структуры проекта (рисунок 17);
- Визуализация диаграмм (рисунок 18);
- Блокировка объектов описания архитектуры на основании утвержденных объектов в ЛОЦМАН:PLM (рисунок 19);
- Работа с функциональными номерами (рисунок 20).

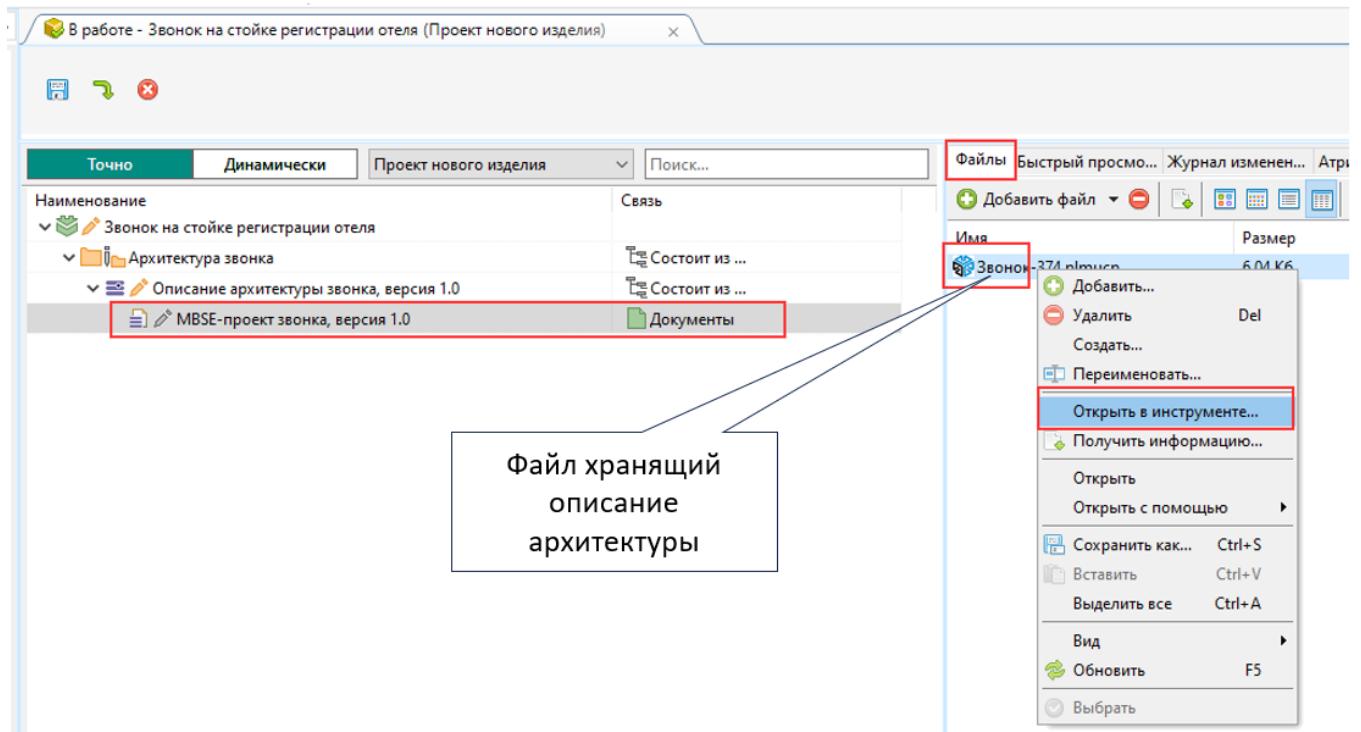


Рисунок 14– Пример открытия файла, хранящего описание архитектуры из ЛОЦМАН:PLM

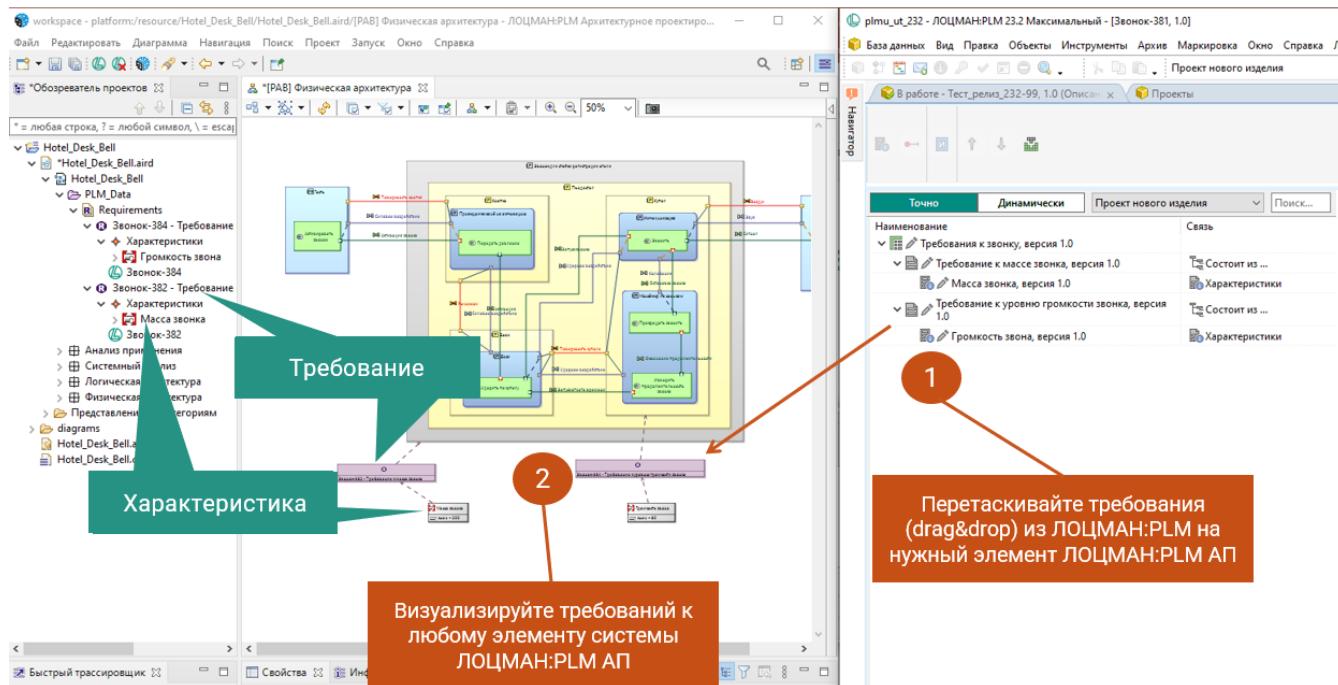


Рисунок 15– Пример выполнения трассировки требований из ЛОЦМАН:PLM на объекты ЛОЦМАН:PLMAP

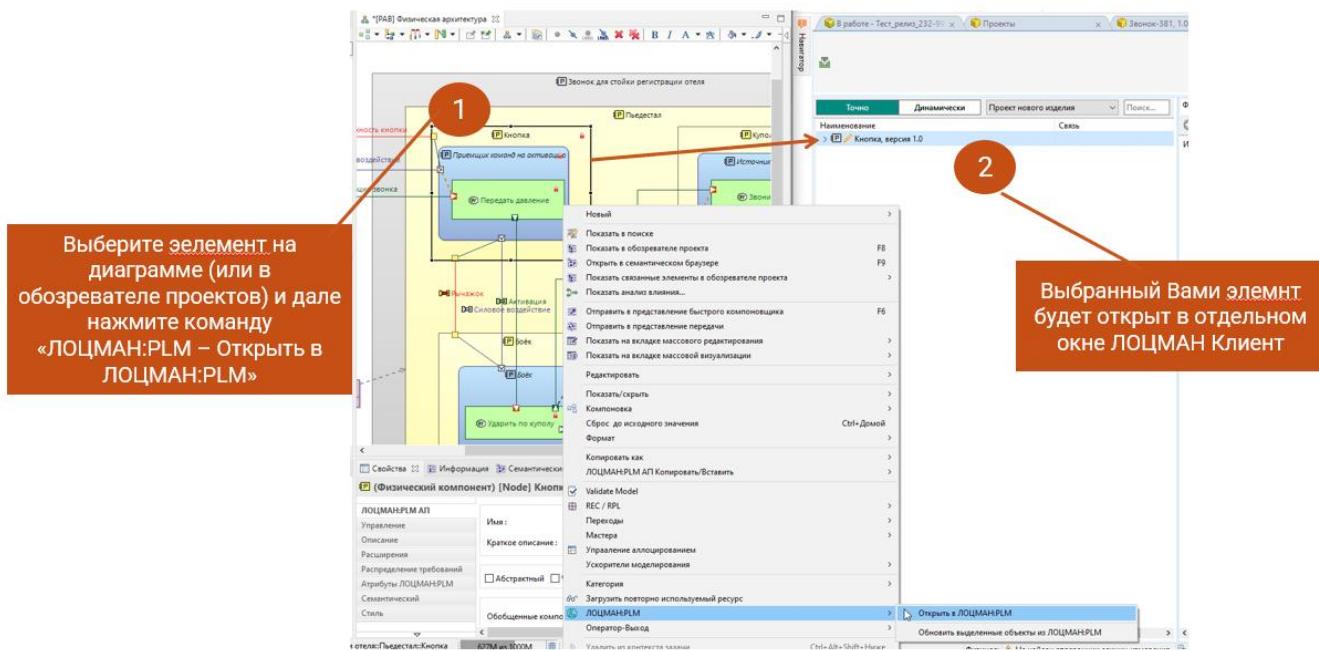


Рисунок 16 - Пример открытия требований из ЛОЦМАН:PLMAP в ЛОЦМАН:PLM

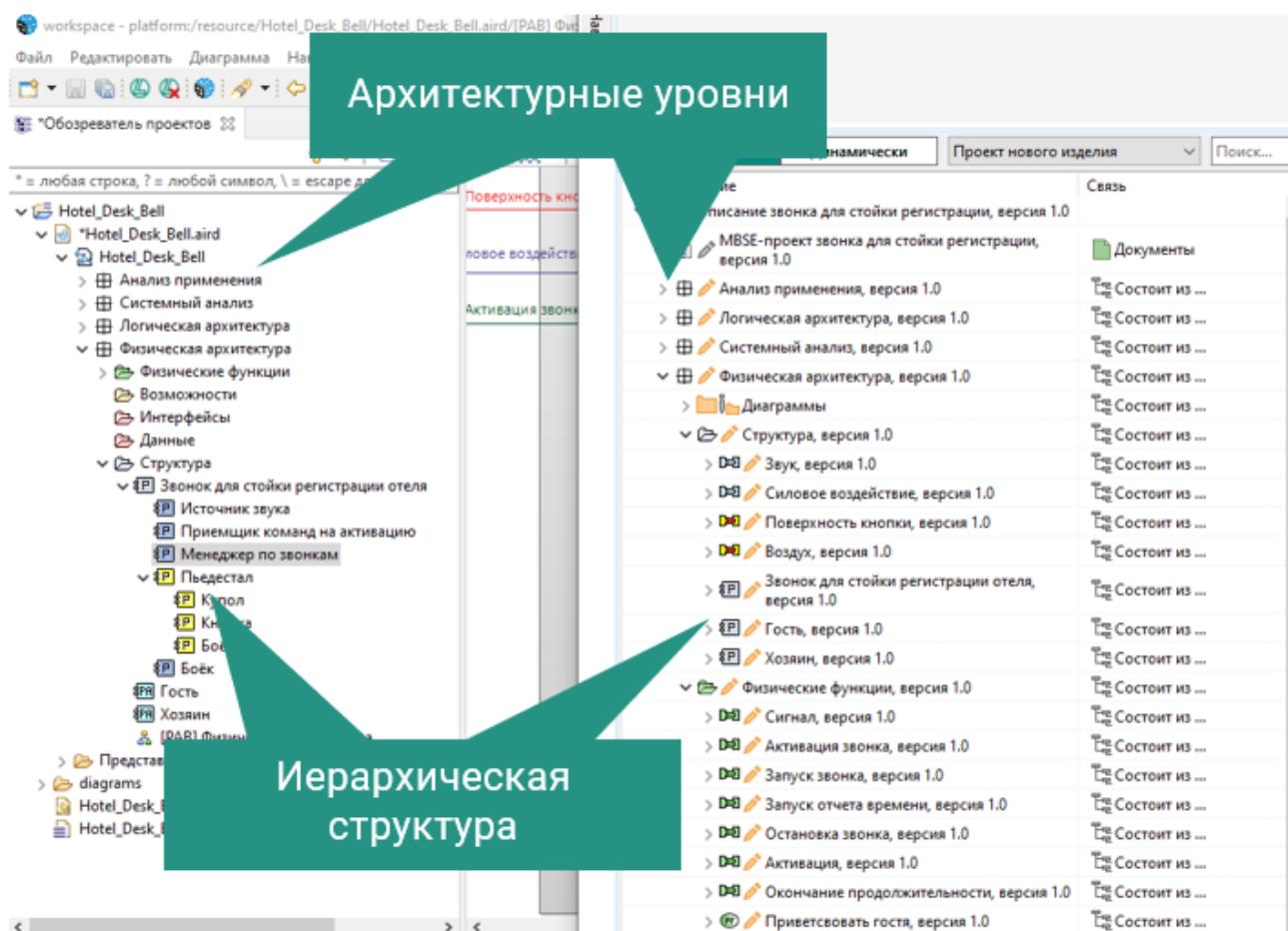


Рисунок 17 - Пример отображения структуры проекта ЛОЦМАН:PLM АП в ЛОЦМАН:PLM

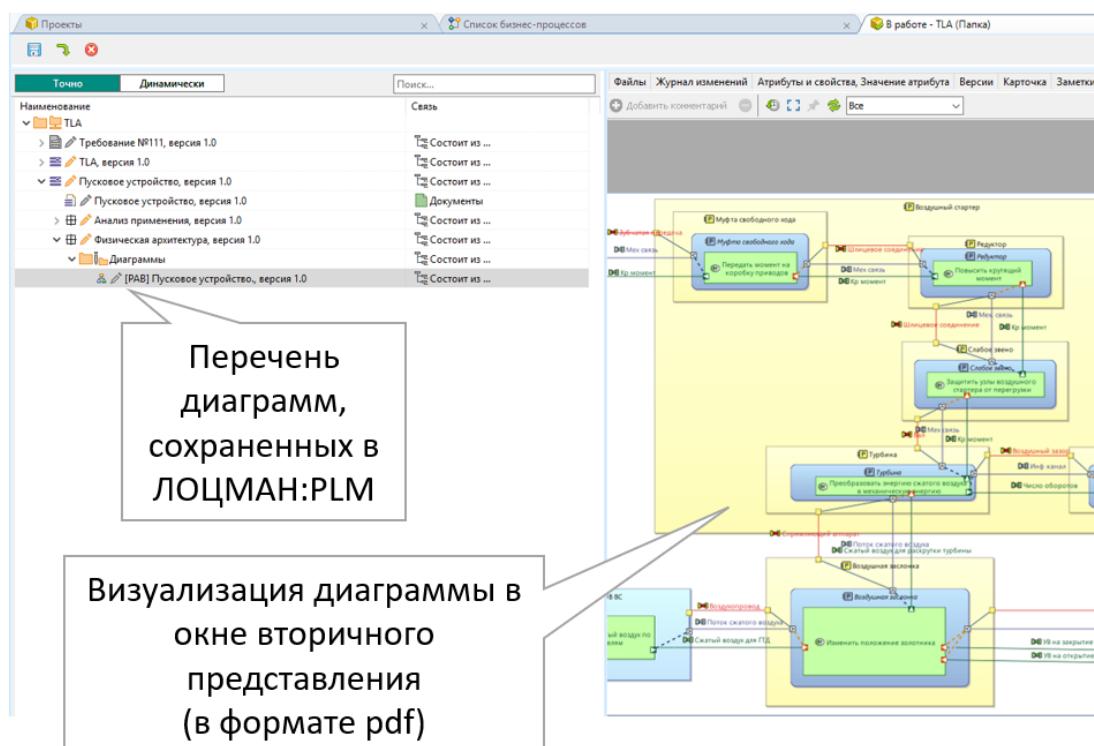


Рисунок 18 – Пример визуализации (разработанных вЛОЦМАН:PLM АП) диаграмм в ЛОЦМАН:PLM

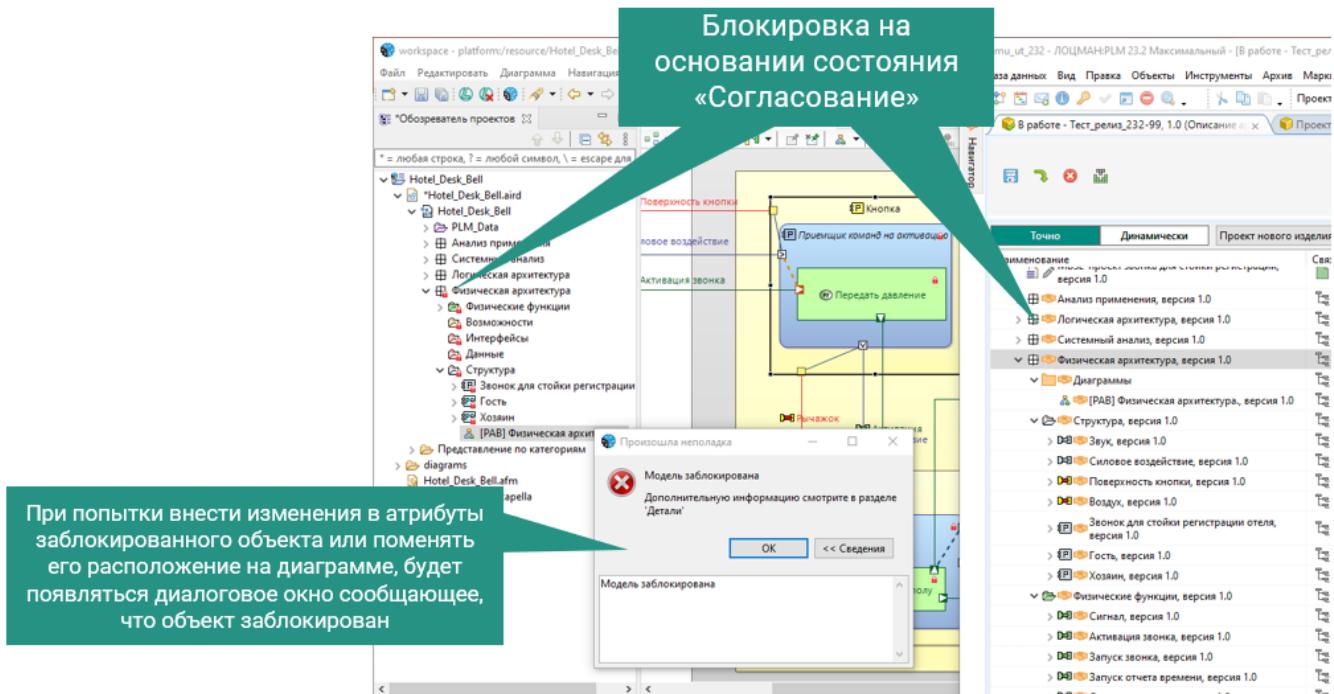


Рисунок 19 - Пример визуализации блокировки элементов модели

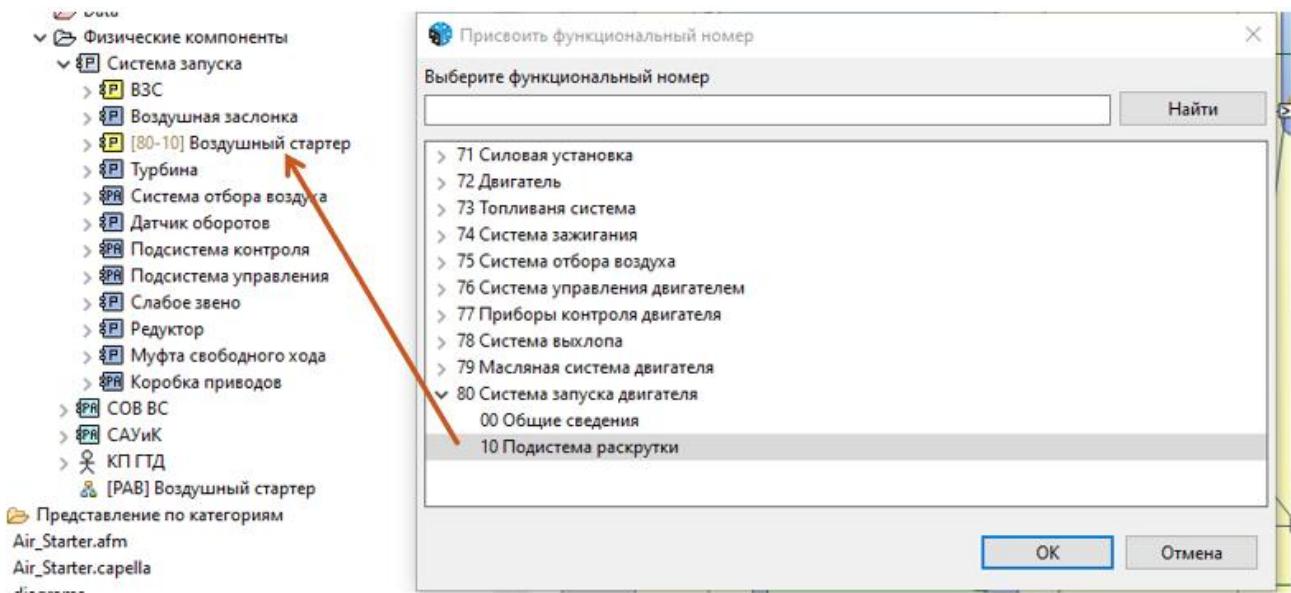


Рисунок 20 - Пример работы с функциональными номерами

3 Правила установки

Установка ПО ЛОЦМАН:PLMAрхитектурное проектирование должна осуществляться в соответствии с инструкцией по установке.

4 Правила эксплуатации

Эксплуатация ПО ЛОЦМАН:PLM Архитектурное проектирование должна осуществляться в соответствии с руководством пользователя.

5 Программные компоненты и совместимость

ЛОЦМАН:PLM Архитектурное проектирование основано на стороннем ПО, перечисленном в этом разделе. Эти компоненты являются программным обеспечением с открытым исходным кодом, лицензированным по лицензии, одобренной Open Source Initiative (OSI, www.opensource.org) или аналогичным лицензиям.

	1	2
Хост	eclipse foundation	eclipse foundation
Компонент	Capella	RequirementsViewpoint
Мин. версия	6.1.0	0.13.0
Использование	Среда системного моделирования, поддерживающая метод Arcadia	Дополнение для работы с требованиями
Лицензия	EPL	EPL
Источник	https://www.eclipse.org/capella/download.html	https://github.com/eclipse-capella/capella-requirements-vp/releases
Домашняя страница	https://www.eclipse.org/capella/	https://mbse-capella.org/addons.html

6 Технологические стеки (компоненты)

ПО включает следующие технологические стеки (компоненты):

1. Delphi.
2. Eclipse SDK.
3. GIT.
4. InnoSetup.
5. Java.