АРХИТЕКТУРНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОГРАММНОЙ ПЛАТФОРМЫ АСУ ТП АЭС ДЛЯ ОБЕСПЕЧЕНИЯ ЖИВУЧЕСТИ

Дагаев Дмитрий Викторович, Генеральный директор ООО «СКАДИ»,
Консультант проекта Информатика-21

Уклон в ИБ или уклон в ФБ? Оба хуже…

Информационная безопасность security

- Средства защиты информации от НСД;
- Модули доверенной загрузки;
- Системы обнаружения вторжений;
- Системы антивирусной защиты;
- Средства криптографической защиты информации.
- Средств защиты от недостаточной ФБ – НЕТ.

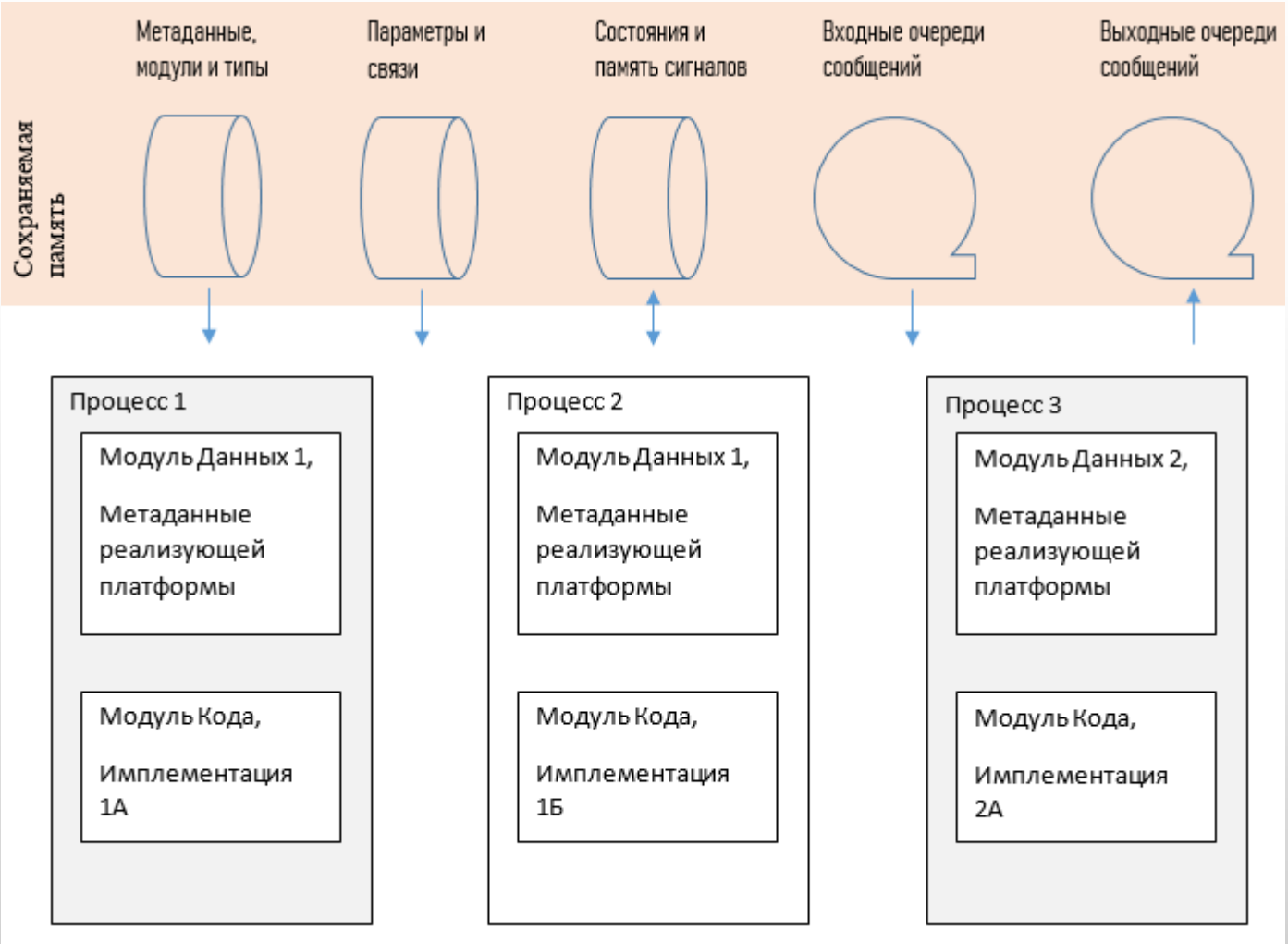
Функциональная безопасность safety

- Средства ограничения функциональности (предельный случай – жесткая логика);
- Принцип единичного отказа и резервирование;
- Диверсификация или принцип разнообразия;
- Сохранение и восстановление состояний;
- Исключение взаимовлияния.

ФБ и тренд от мейнфреймов к клеточной структуре для повышения живучести

- Отказоустойчивость и резервирование: от принципа единичного отказа к живучести при многих, известны системы до 13 уровней *элегантной деградации*;
- Диверсификация или *принцип разнообразия*;
- Сохранение в восстановление *состояний*;
- Ремонтопригодность и возможность *горячей замены*;
- Исключение взаимовлияния компонентов;
- Самодиагностика в реальном времени;
- *Эргодичность* и уменьшение старения клеток;
- *Ограничение сложности* ПО.

Программирование, управляемое данными DOP

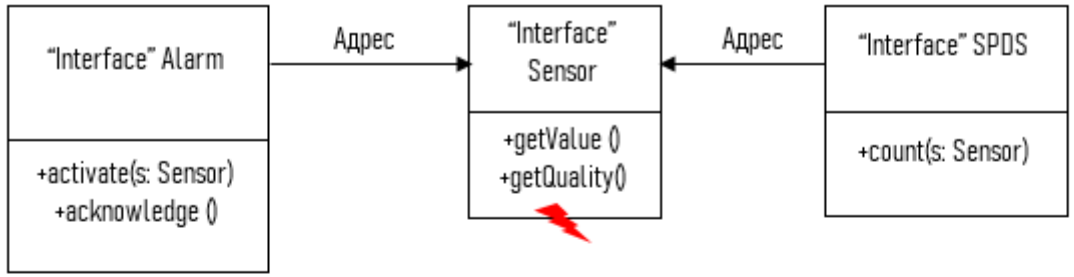


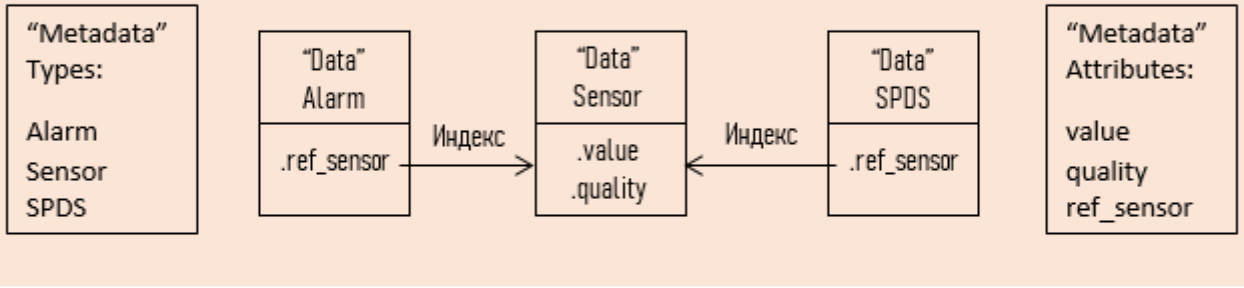
Принципы DOP Евгения Кузнецова (IBM) упоминаемого как автора, на которых реализован ряд дата-центричных технологий.

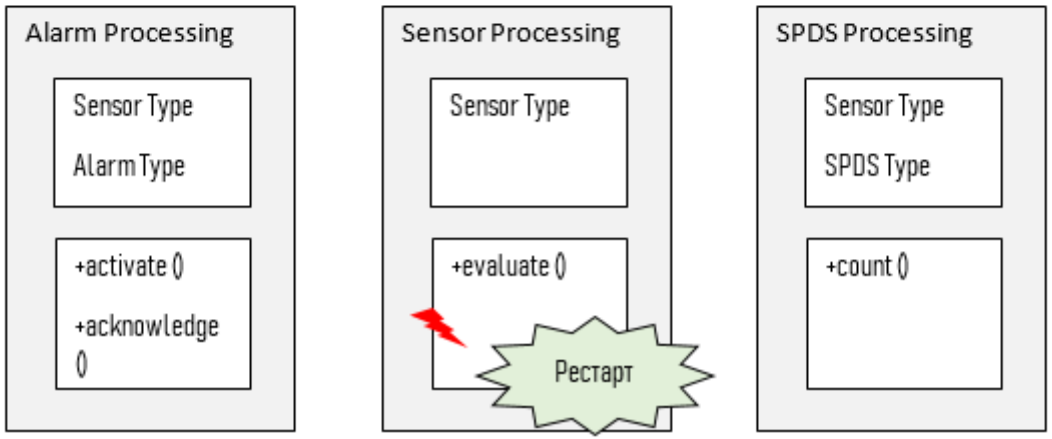
- Сепарация кода от данных;
- Предоставление данных и метаданных на основании прав доступа вместо инкапсуляции;
- Обработка данных на основе метаданных реализующей платформы;
- Скрытие подробностей кода.

Реализация средствами модульного программирования Оберон.

Инвариант связей, изоляция и рестарт







Обработка ошибок адресуемого объекта в OOP (сверху) сопряжена с необходимостью изоляции ошибки, рестарта и восстановления связей по адресам.

Обработка ошибок модуля в DOP:

- Отключение обработчика с установкой недостоверного качества;
- Рестарта модуля или изолированного процесса с сохранением/восстановлением состояния в памяти;
- Инвариант связей – персистентные индексные связи между данными;
- Метаданные также являются персистентными.

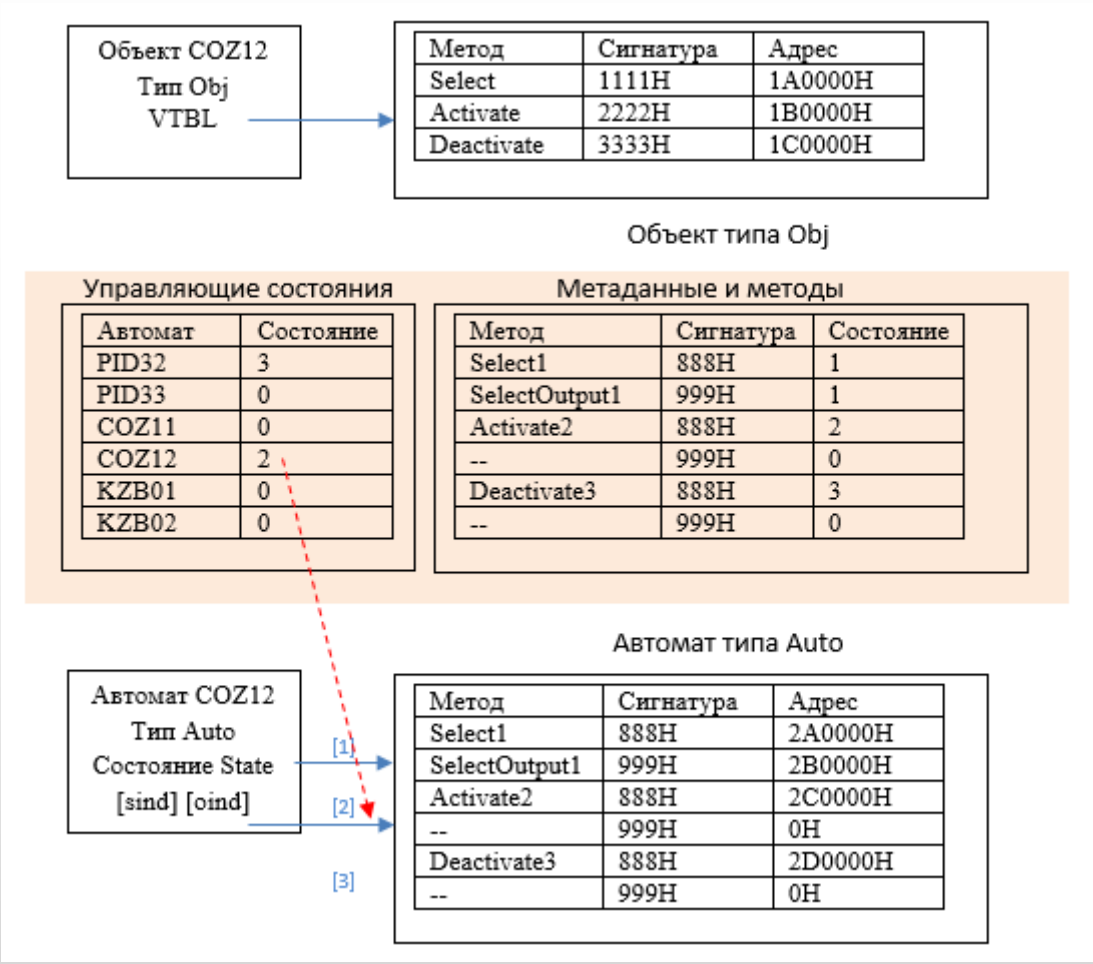
Парадигма автоматного программирования (А.А.Шалыто)

Основные пункты концепции:

- Программы предлагается создавать так же, как производится автоматизация технологических (и не только) процессов;
- Основа – выделение управляющих состояний;
- Верификация автоматных программ.

Разработка автором исполняющей машины автоматных программ, сопряжение с дата-центрической технологией и модульным программированием.

Инвариант состояний автомата



1. В отличие от таблицы методов ООР таблица методов DOP индексируется состояниями;
2. Метаданные и управляющие состояния сохраняются в долговременной памяти;
3. Метаданные реализующей платформы восстанавливаются при запуске ПО;
4. Для состояния N исполняется метод перехода «ПереходN» и метод выхода «ВыходN»;
5. Состояние переключается исполняющей машиной автоматных программ.

Условия горячей замены

Принцип подстановки Лисков: *Функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом.*

Подстановка без детализации структуры – не одно и то же: Compiler.Start() <> C400.Start()

Принцип безопасной подстановки – инвариант, сохраняющий

X, Y, Z, δ, φ, y0, F для автомата в долговременной памяти, где

X-входы, Y-состояния, Z-выходы, δ-сигнатура функции перехода, φ-сигнатура функции выхода, y0-начальное состояние, F-допускающие состояния. Обеспечивается инвариант графа переходов.

Принцип безопасной подстановки используется как условие горячей замены, рестарта, восстановления.

Система семантических ограничений

Проблема сохранения состояния рекурсивных процедур;

Проблема дублирования структур данных во внутренней памяти;

Проблема ограничения интервала расчета.

Запрет рекурсии

```
RESTRICT –PROCEDURE(PROCEDURE);
```

Запрет динамической памяти

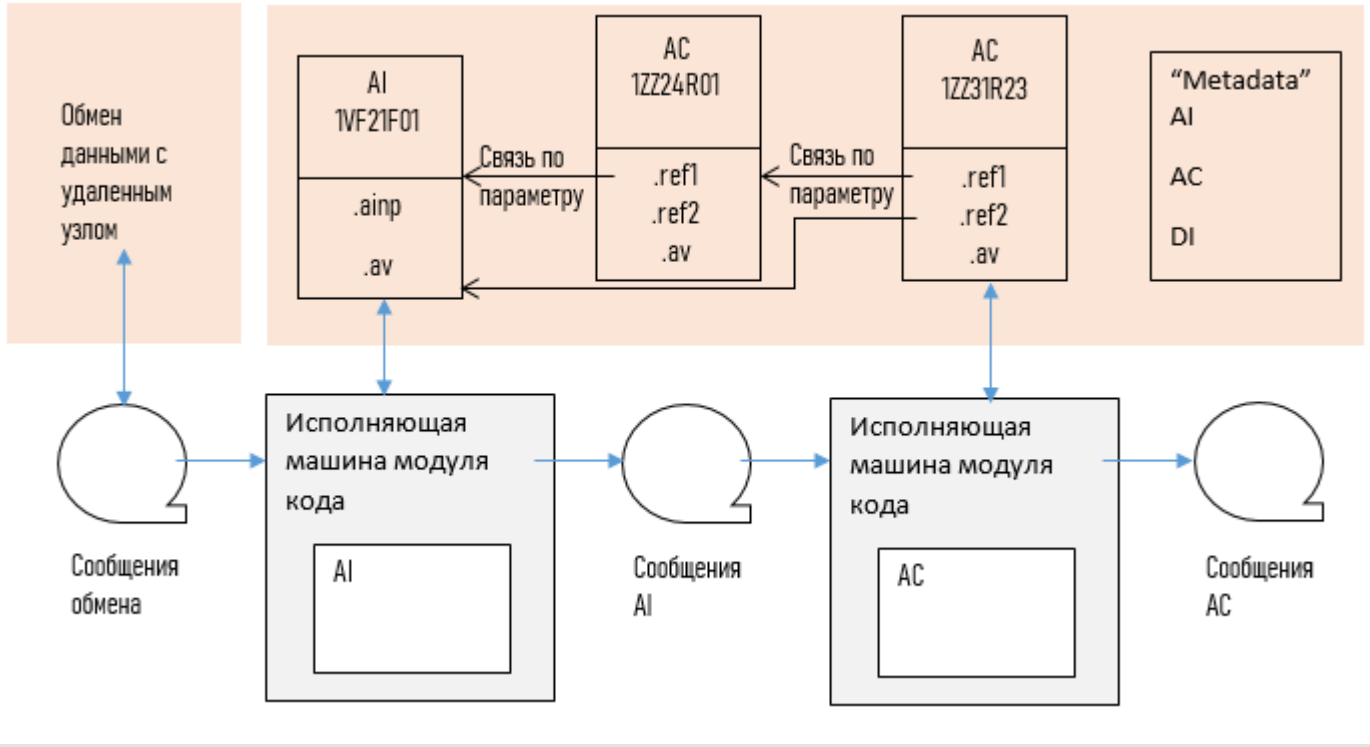
```
RESTRICT –NEW, -POINTER, -SYSTEM;
```

Ограничение итераций цикла

```
RESTRICT –WHILE, -REPEAT, -UNTIL,
-LOOP;
```

Система МультиОберон со сменными бэкендами Native, C, LLVM доступна по https://github.com/dvdagaev/Mob. Текущая версия 1.2 поддерживает архитектуры X86, X64, ARM32, Aarch64 для Linux, Windows, Raspberry Pi OSa.

Метод сборочного программирования (Е.М.Лаврищева, В.Н.Грищенко)



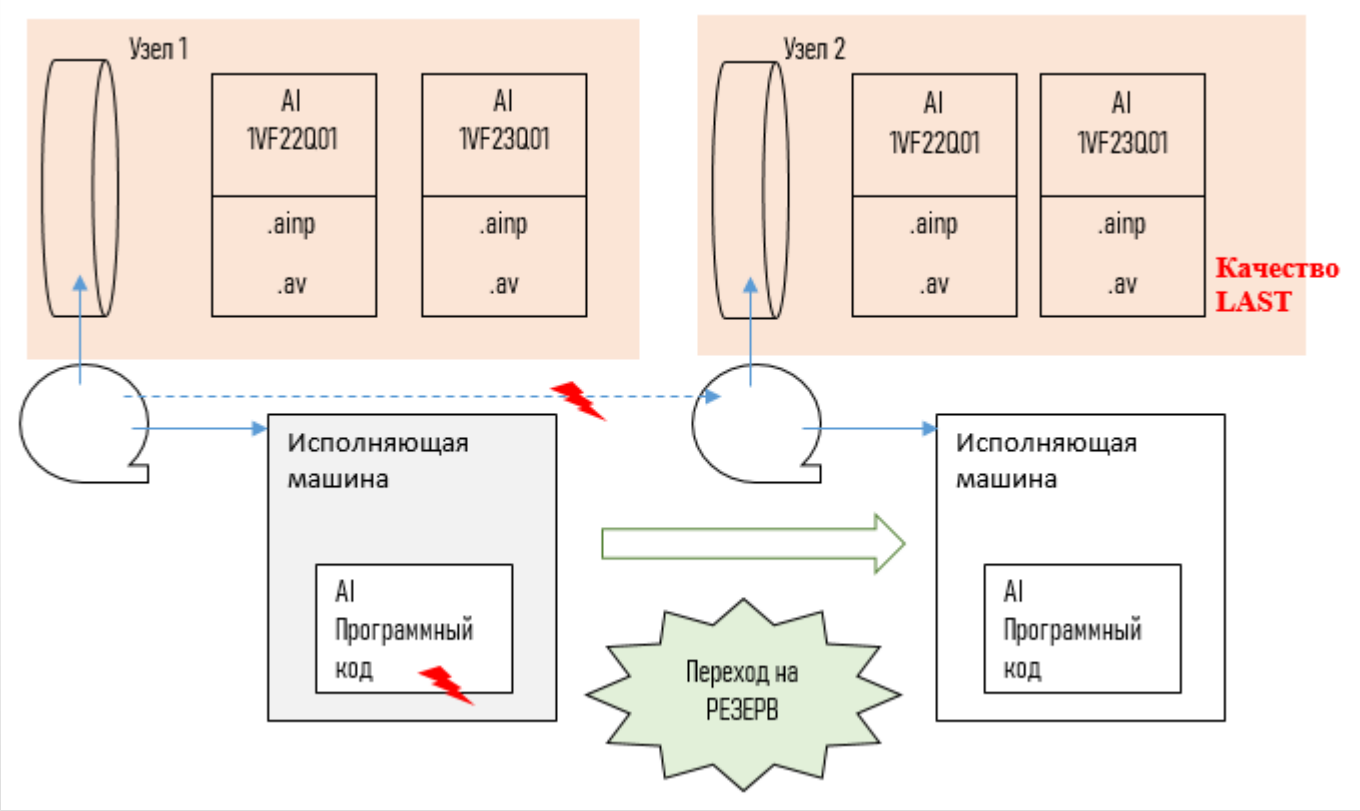
Информационная связь, как обмен данными между модулями: сообщения + связи по параметру

Вырисовывается схема сборки: объекты и связи.

Свойства объекта-модуля:

1. Обособленные данные в памяти;
2. Динамически заменяемый модуль кода;
3. Управление с помощью исполняющей машины ИМАП;
4. Параметрические связи и метаданные;
5. Раздельная компиляция Oberon;
6. Динамическая загрузка в различных точках;
7. Унификация механизмов обмена и управления.

Инвариант значений и синхронизация



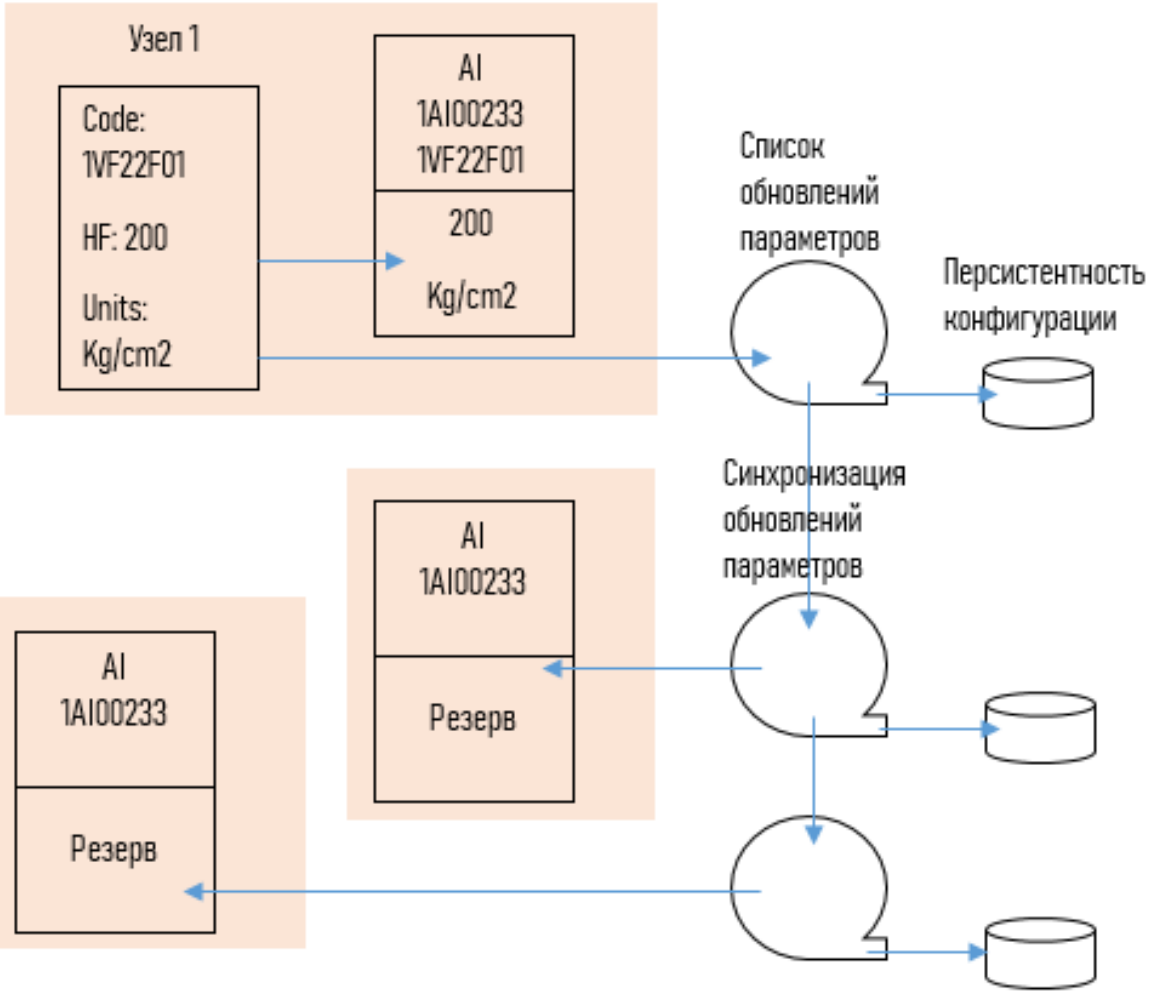
Значения сигналов синхронизируются в узлах распределенной памяти.

Механизм синхронизации использует очереди сообщений и коммуникационное ПО.

Инвариант значений обеспечивается системой синхронизации и с проверкой целостности. Качество LAST при фиксации проблем.

Динамическая реконфигурация (переход на РЕЗЕРВ). Восстановление синхронизации.

Инвариант параметров и персистентность



Изменение конфигурации сборки в режиме онлайн.

Изменения параметров являются персистентными и сохраняют свои значения при возобновлении питания.

Инвариант параметров обеспечивается проверкой целостности на узлах на протяжении всего времени жизни системы.

Эшелонированный доступ к данным

Предоставление доступа к данным DOP осуществляется многоуровневой системой.

- Первый уровень доступа к сегменту подсети обеспечивается информационными диодами.
- Второй уровень доступа к файлам/сегментам разделяемой памяти осуществляется средствами ОС Astra Linux.
- Третий уровень доступа к отдельным полям данных и метаданных только по чтению осуществляется средствами компилятора языка Оберон.

Какое отношение имеем к ОС?

- Тенденция: (1 Комп + Много пользователей) -> (1 Комп + 1 Пользователь) -> (Много компьютеров + 1 пользователь);
- *Фазовый переход* – распределенная система из редко обслуживаемых узлов (От OSDAY к CELLDAY?);
- Свойства функциональной безопасности, заложенные в архитектуру;
- Переход к более простым решениям (Хоар: Простое ПО без недостатков vs Сложное без очевидных недостатков).

На данных архитектурных принципах реализовано ПО «СКАДИ» для АСУ ТП АЭС.

Дагаев Дмитрий Викторович dvdagaev@oberon.org

Вопросы по докладу …

Приложение1: Принцип единичного отказа – Ariane 5 Flight 501

Отказ основного и резервного компьютера, приведший к потере спутника 4 июня 1996. Стоимость программы $8 billion, полезная нагрузка спутника $500 million.

“The greater horizontal acceleration caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to overflow and cause a hardware exception. ***Efficiency considerations had omitted range checks for this particular variable***, though conversions of other variables in the code were protected. The exception halted the reference platforms, resulting in the destruction of the flight.”

Преобразование данных из 64-битного действительного числа в 16-битное целое привело к переполнению и вызвало аппаратное исключение. ***По соображениям эффективности были опущены проверки диапазонов*** для данной конкретной переменной.

Программный код на Ада:

```
P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S ((1.0/C_M_LSB_BH) *
G_M_INFO_DERIVE(T_ALG.E_BH)));
```