Инструменты автоматизации процессов разработки и сертификации интегрированной модульной авионики

Хорошилов Алексей
khoroshilov@ispras.ru

ISP RAS

План

- Архитектура интегрированной модульной авионики (ИМА)
- Процесс разработки ИМА
- Автоматизация процессов разработки ИМА
  - Системное интеграция
  - Системное тестирование

Авионика

- Avionics = "aviation" + "electronics”
- БРЭО = Бортовое радиэлектронное оборудование



Авиационные функции



датчик1



датчик2



актуатор1



актуатор2

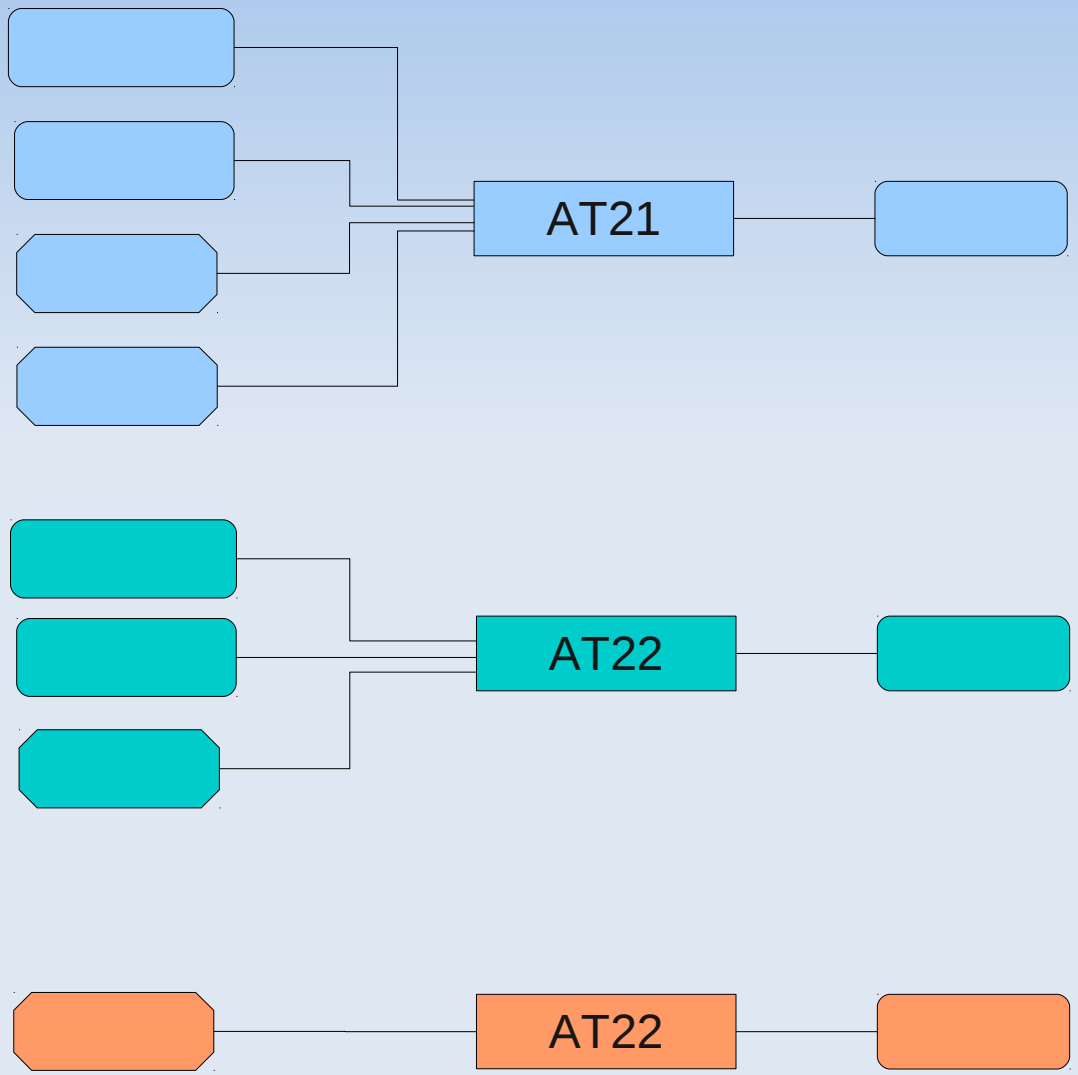
Управляющий Модуль



панель управления

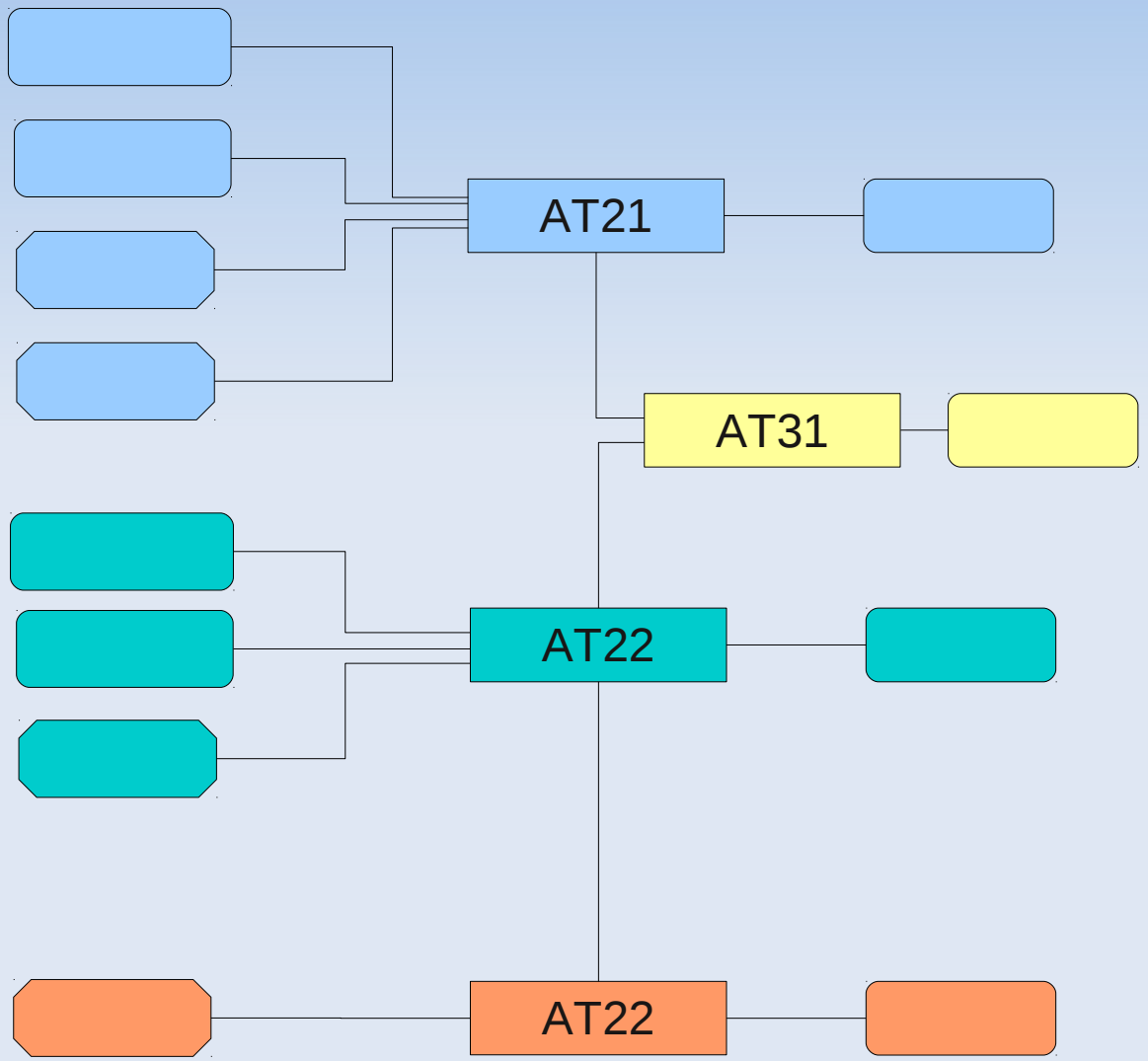


Архитектура первого поколения



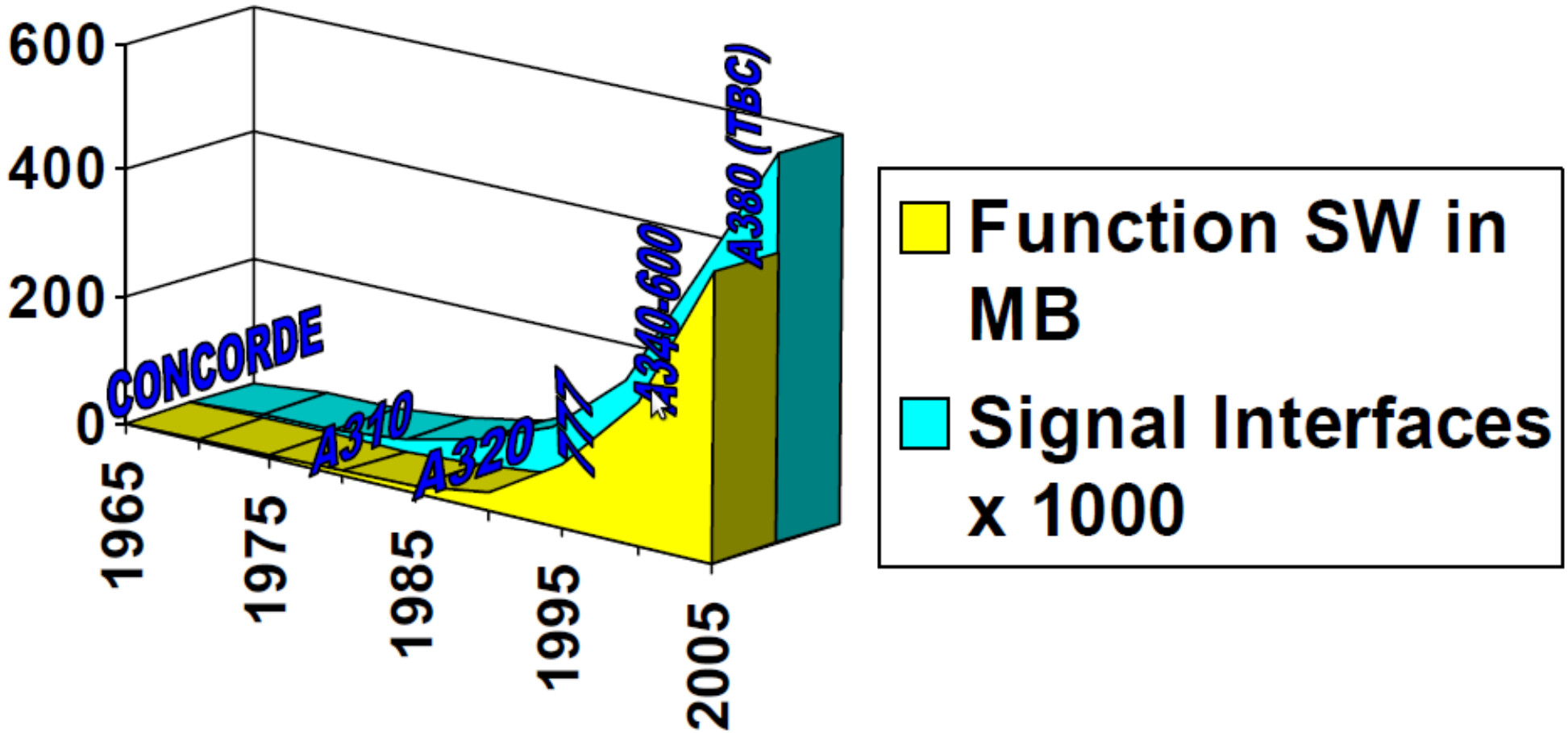
- Каждая функция – полностью независимая система

Федеративная архитектура



- Каждая функция – свой компьютер
- Каждое соединение – выделенный провод

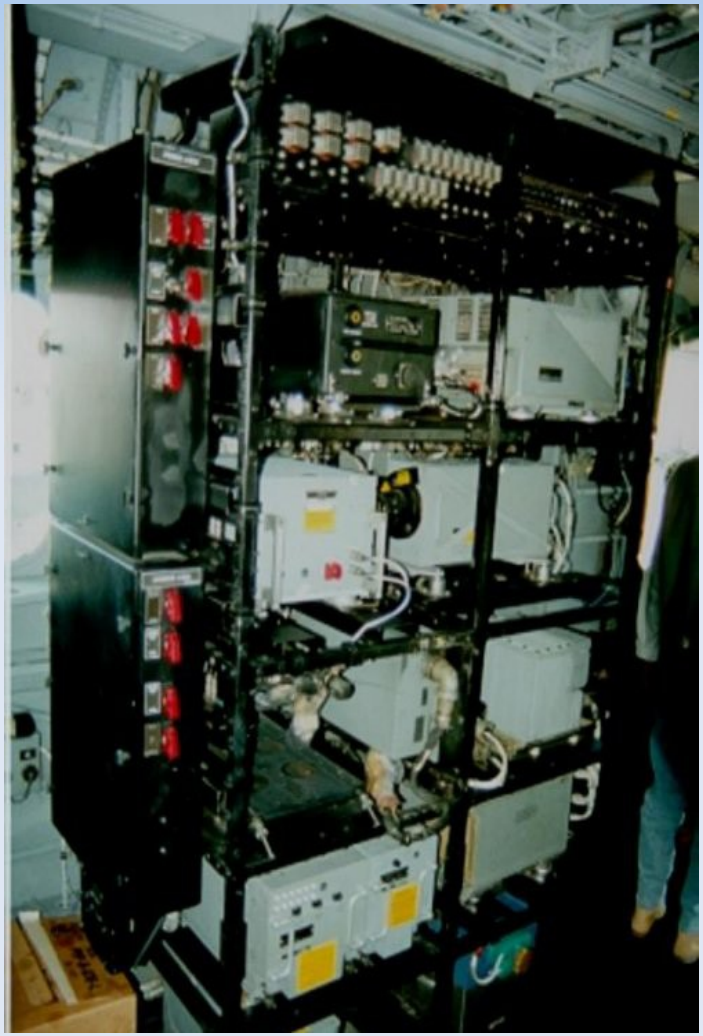
Рост применения авионики (1)



Федеративная архитектура (2)



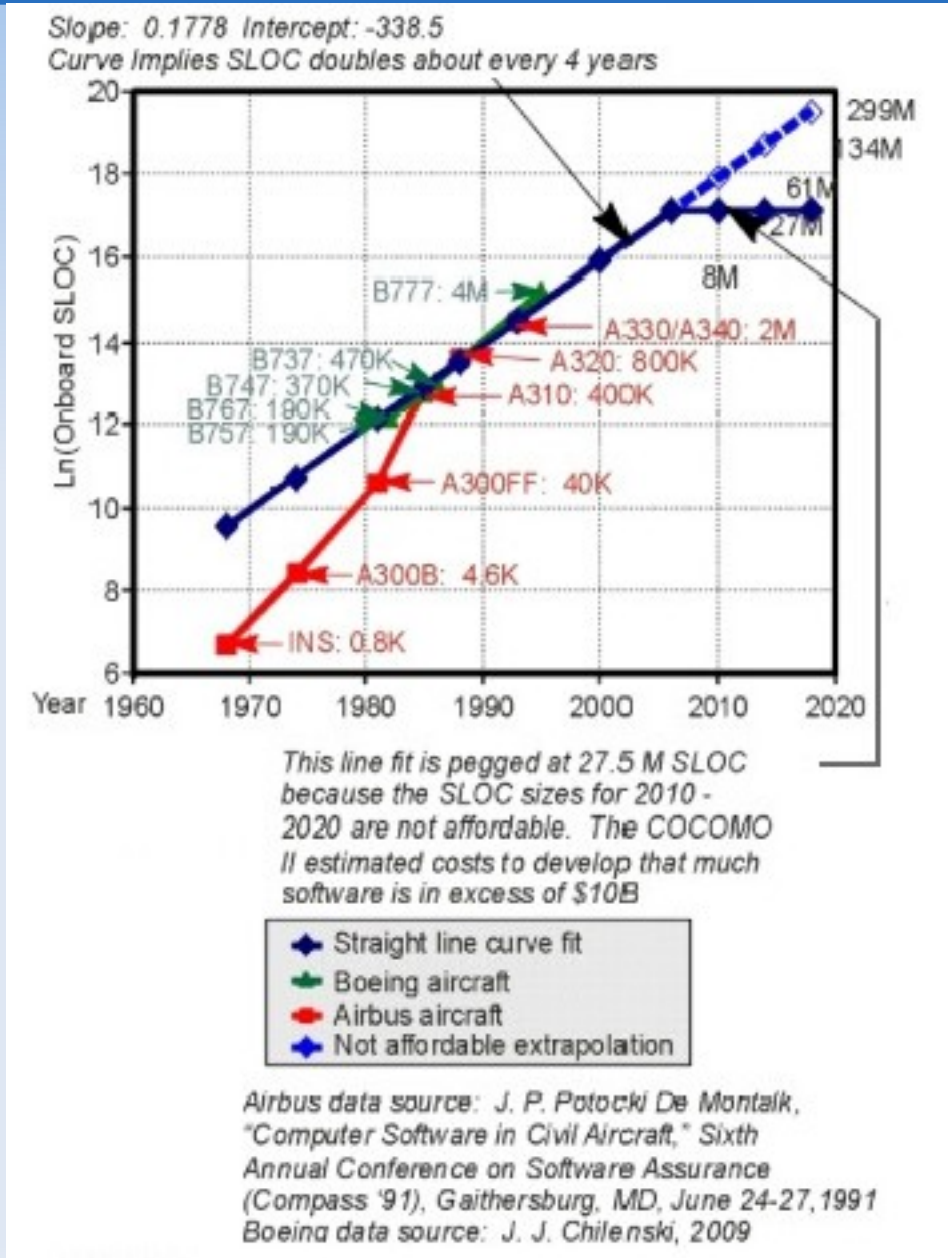
Федеративная архитектура (3)





Рост применения авионики (2)

- >100км проводов на 1 самолете
- Несколько тонн веса
- Энергопотребление



P. Feiler, L. Wrage, J. Hansson, System Architecture Virtual Integration: A Case Study Embedded Real-time Software and Systems Conference (ERTS2010), May 2010.

Интегрированная модульная авионика

- Виртуализация:
  - На логическом уровне ничего не изменилось, а на физическом кардинальные изменения

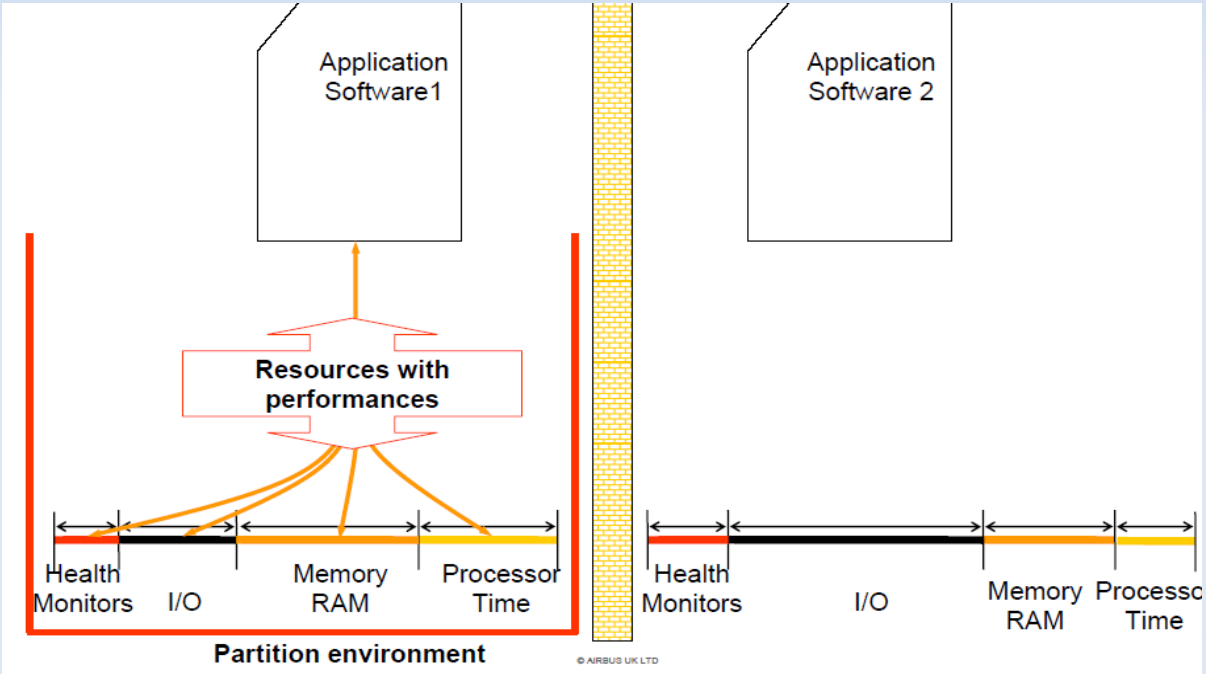
Интегрированная модульная авионика (1)

- Унификация
- ”Виртуализация” блоков питания



Интегрированная модульная авионика (2)

- ”Виртуализация” процессорных блоков
- ”Виртуализация” проводов - ARINC-429

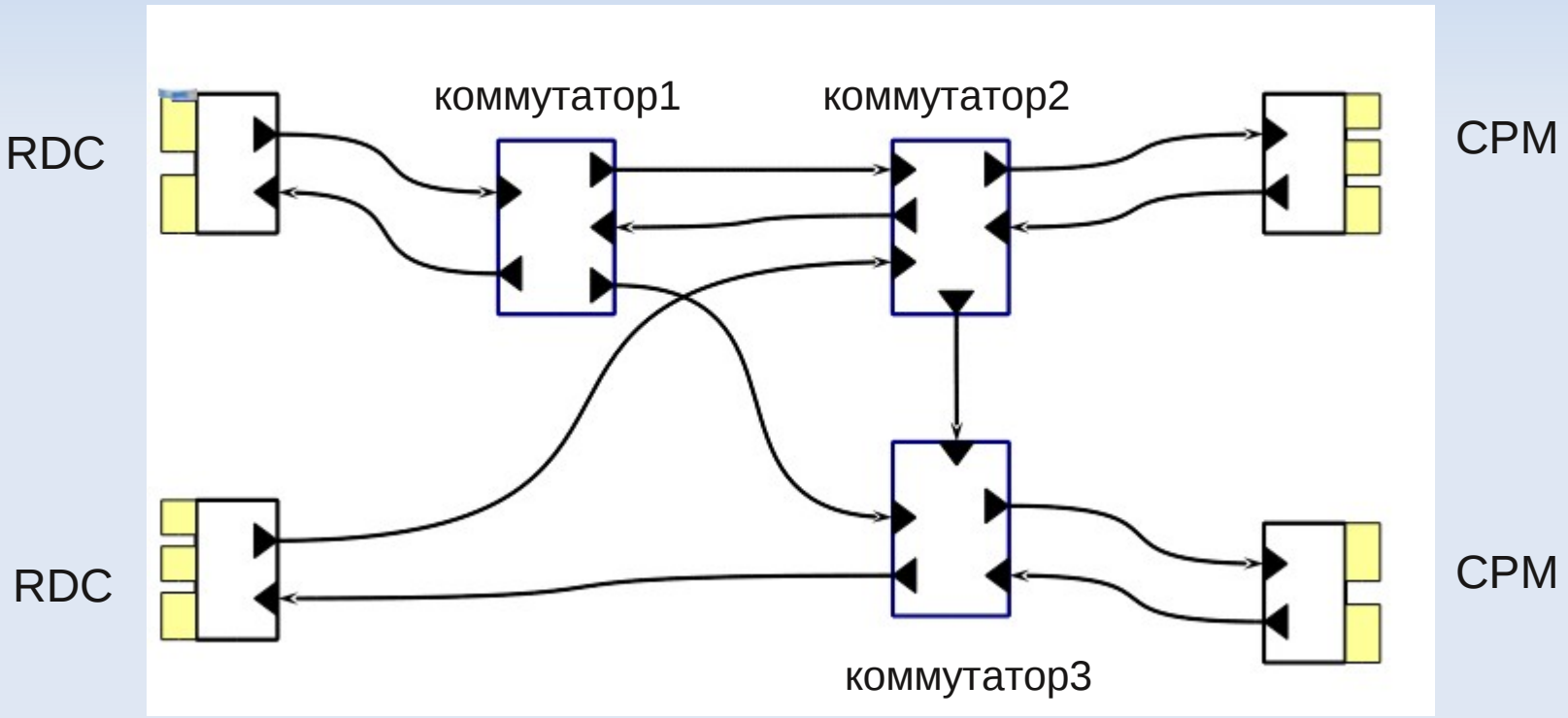


ARINC-653 Operation System

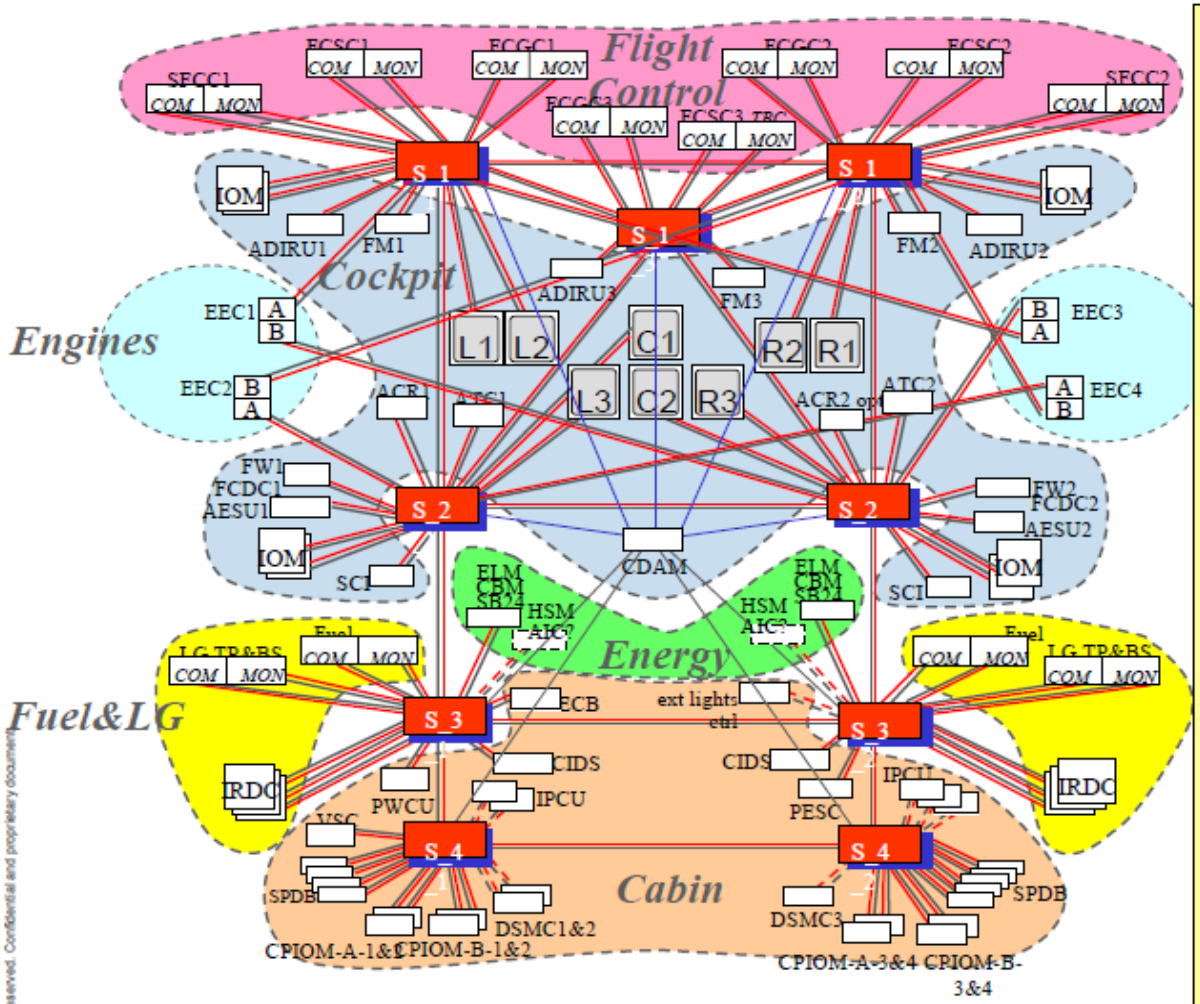


Интегрированная модульная авионика (3)

- ”Виртуализация” сети:
  Avionics Full-Duplex Switched Ethernet



ИМА на Airbus-380



AFDX Network:

- 100 Mbits
- Redundant Network (A&B) with independent alimentation
- AFDX switches = 2 x 8
- NB of ports (connections) possible on each switch (20-24)
- MTBF of the switch is very high (100 000 hours expected)
- Up 80 AFDX subscriber

© AIRBUS UK LTD. All rights reserved. Confidential and proprietary document

ИМА в России

- SUKHOI SUPERJET-100



План

- Архитектура интегрированной модульной авионики (ИМА)
- **Процесс разработки ИМА**
- Автоматизация процессов разработки ИМА
  - Системное интеграция
  - Системное тестирование

Разработка авионики

регламетируется:

- ARP-4754 Руководство по рзработке высокоинтегрированных систем
- ARP-4761 Методы оценки безопасности
- DO-254 Руководство по обеспечению проектирования электронного аппаратного оборудования самолета
- DO-178B Требования к ПО бортовой аппаратуры и систем при сертификации авиационной техники

Разработка авионики: федеративная архитектура

- Поставщик авиационной функции разрабатывает и интегрирует
  - аппаратуру (CPM, датчики, актуаторы)
  - ОС
  - прикладное ПО
- Системный интегратор
  - интегрирует функции между собой
  - проводит системное и интеграционное тестирование

Разработка авионики: IMA

- Поставщик платформы разрабатывает
  - аппаратные модули (CPM, коммутаторы, RDC)
  - ОС
- Поставщик авиационной функции разрабатывает
  - специфическую аппаратуру (датчики, актуаторы)
  - прикладное ПО
- Системный интегратор
  - конфигурирует и верифицирует CPM и AFDX
  - проводит системное и интеграционное тестирование

План

- Архитектура интегрированной модульной авионики (ИМА)
- Процесс разработки ИМА
- **Автоматизация процессов разработки ИМА**
  - Системное интеграция
  - Системное тестирование

Автоматизация процессов разработки ИМА

совместный проект ГосНИИАС и ИСП РАН, стартовавший в 2010 году

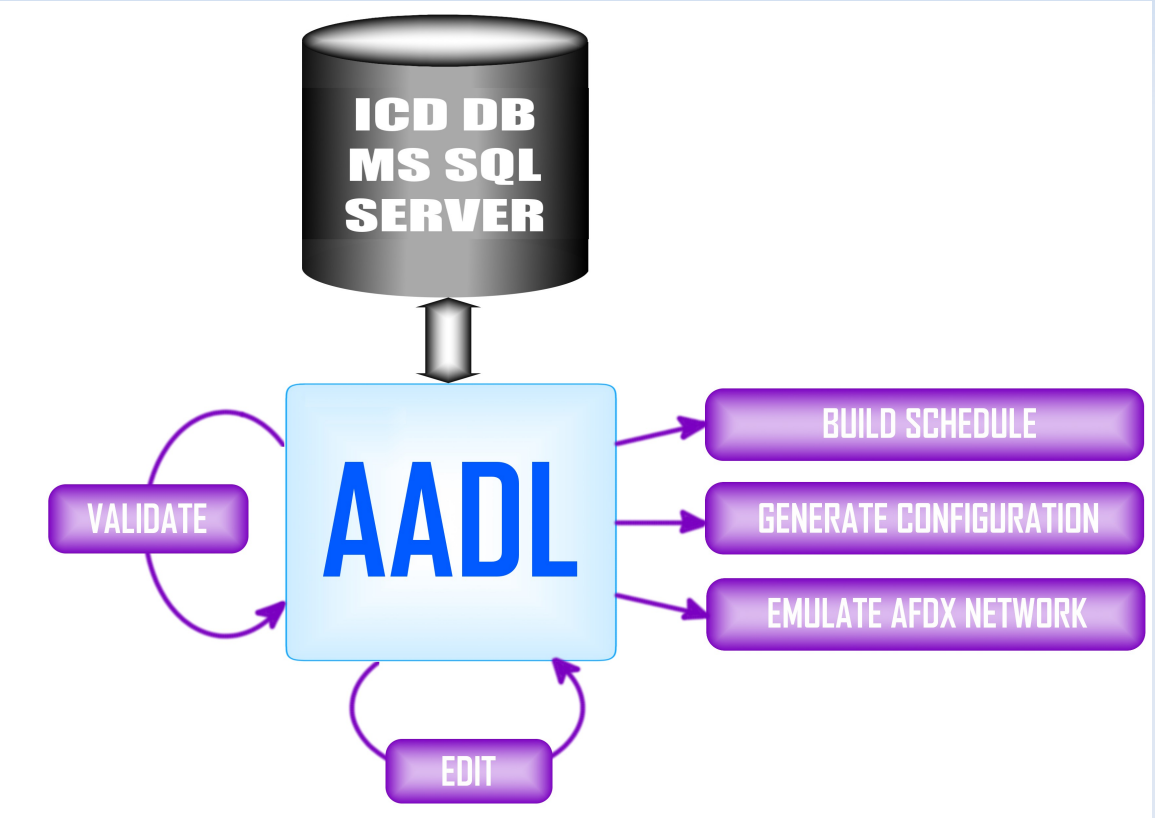
ближайшие планы:

- автоматизация системной интеграции
- автоматизация системного тестирования

АРМ СИ (1)

Model-Driven Development

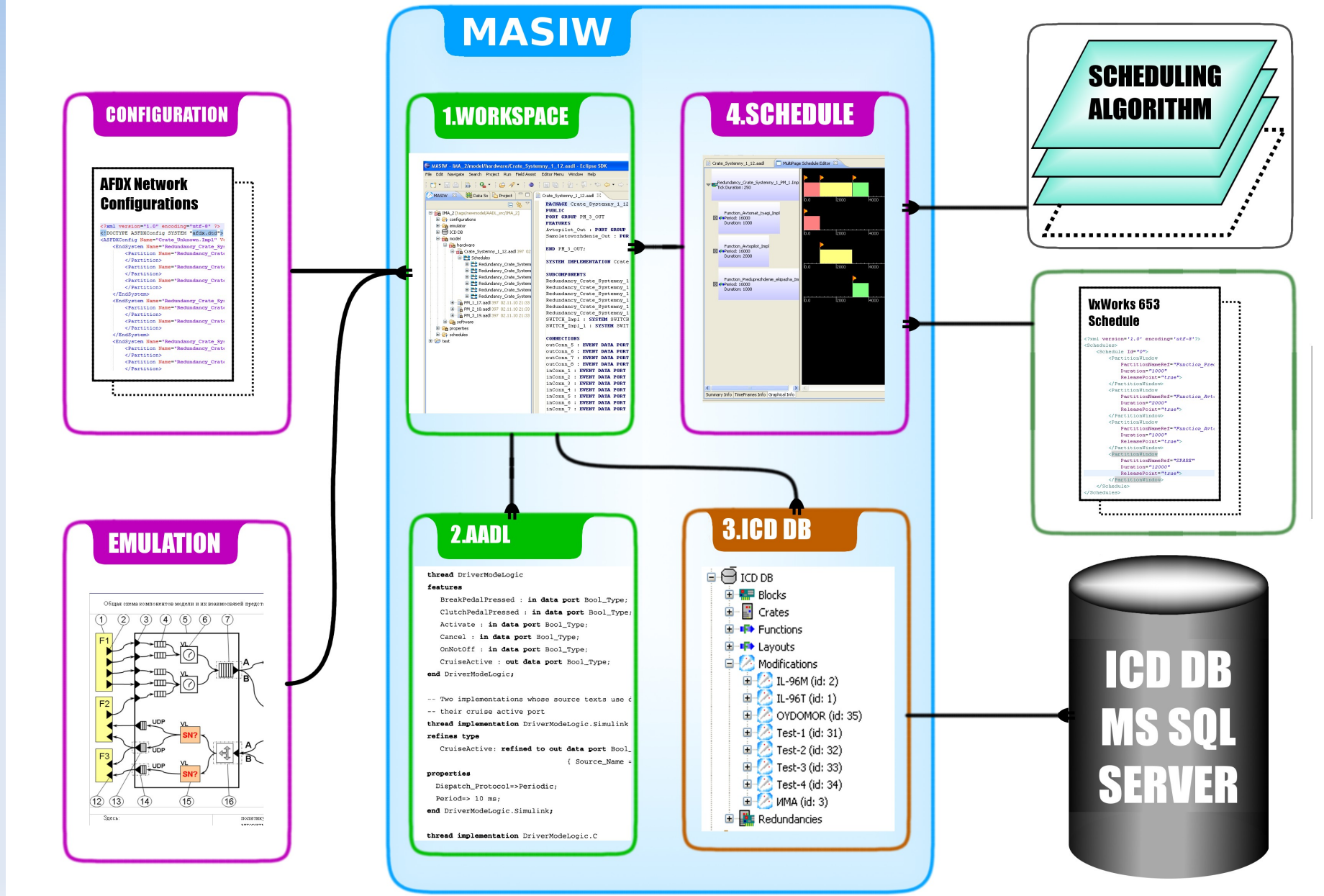
- AADL модель архитектуры авионики



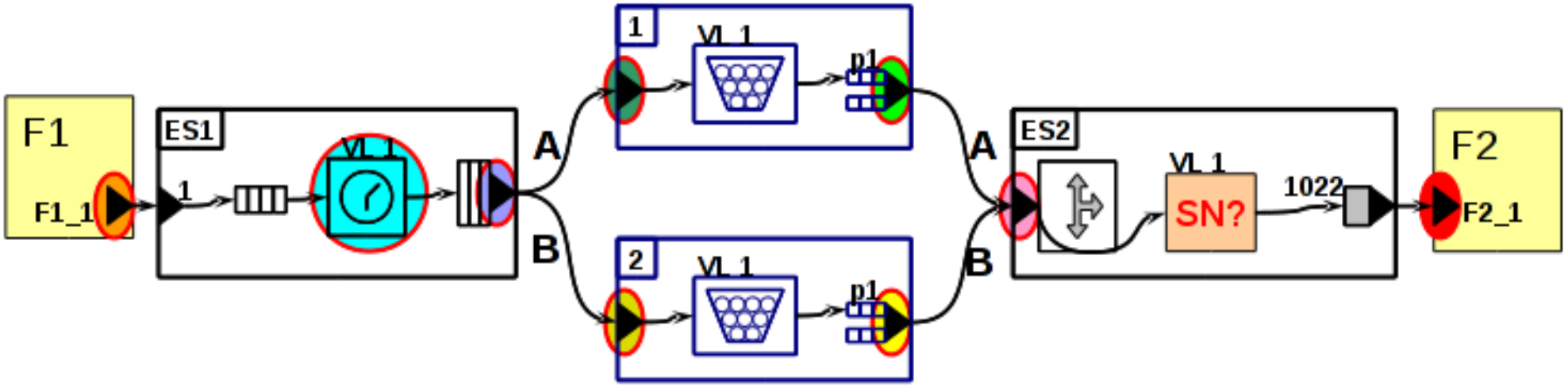
АРМ СИ (2)

- верификация AADL-модели архитектуры
  - совместимость по ресурсам
  - выполнимость ограничений платформы и приложений
- составление расписаний для ARINC-653 и AFDX
- генерация конфигураций для CPM и AFDX-коммутаторов
- оптимизация архитектуры IMA
- исследование характеристик AFDX сетей

АРМ СИ (3)


MASIW
CONFIGURATION
AFDX Network Configurations
1.WORKSPACE
4.SCHEDULE
SCHEDULING ALGORITHM
VxWorks 653 Schedule
EMULATION
2.AADL
thread DriverModeLogic
features
BreakPedalPressed : in data port Bool_Type;
ClutchPedalPressed : in data port Bool_Type;
Activate : in data port Bool_Type;
Cancel : in data port Bool_Type;
OnNotOff : in data port Bool_Type;
CruiseActive : out data port Bool_Type;
end DriverModeLogic;
properties
Dispatch_Protocol=>Periodic;
Period=> 10 ms;
end DriverModeLogic.Simulink;
3.ICD DB
ICD DB
Blocks
Crates
Functions
Layouts
Modifications
IL-96M (id: 2)
IL-96T (id: 1)
OYDOMOR (id: 35)
Test-1 (id: 31)
Test-2 (id: 32)
Test-3 (id: 33)
Test-4 (id: 34)
ИМА (id: 3)
Redundancies
ICD DB
MS SQL
SERVER

АРМ СИ (4)



	0.000: NOTHING: send for VL#1 to ES MUX
	0.000: NOTHING: receive from AFDX#1022 for F2_1@F2@ES2
	0.250: msg [#0 from F1_1@F1@ES1, len=200]: send from F1_1@F1@ES1 to AFDX#1
	1.000: NOTHING: receive from AFDX#1022 for F2_1@F2@ES2
ES1	1.000: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: send for VL#1 to ES MUX
	1.000: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: start send from ES MUX
1	1.021: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: receive by Switch#1
	1.021: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: start send from switch port 'p1'
2	1.021: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: receive by Switch#2
	1.021: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: start send from switch port 'p1'
ES2	1.042: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: receive by de-MUX of ES 'ES2'
	1.042: frame {#0 of [#0 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=0}: receive by de-MUX of ES 'ES2'
	1.250: msg [#1 from F1_1@F1@ES1, len=200]: send from F1_1@F1@ES1 to AFDX#1
	2.000: msg [#0 from F1_1@F1@ES1, len=200]: receive from AFDX#1022 for F2_1@F2@ES2
ES1	2.000: frame {#0 of [#1 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=1}: send for VL#1 to ES MUX
	2.000: frame {#0 of [#1 from F1_1@F1@ES1, len=200], data-len=200, VL=1, SN=1}: start send from ES MUX

АРМ СИ (5)

Ближайшие планы

- верификация AADL-модели архитектуры
- графический AADL-редактор
- развитие AFDX-симулятора
- реализация аналитических методов анализа AFDX-сети

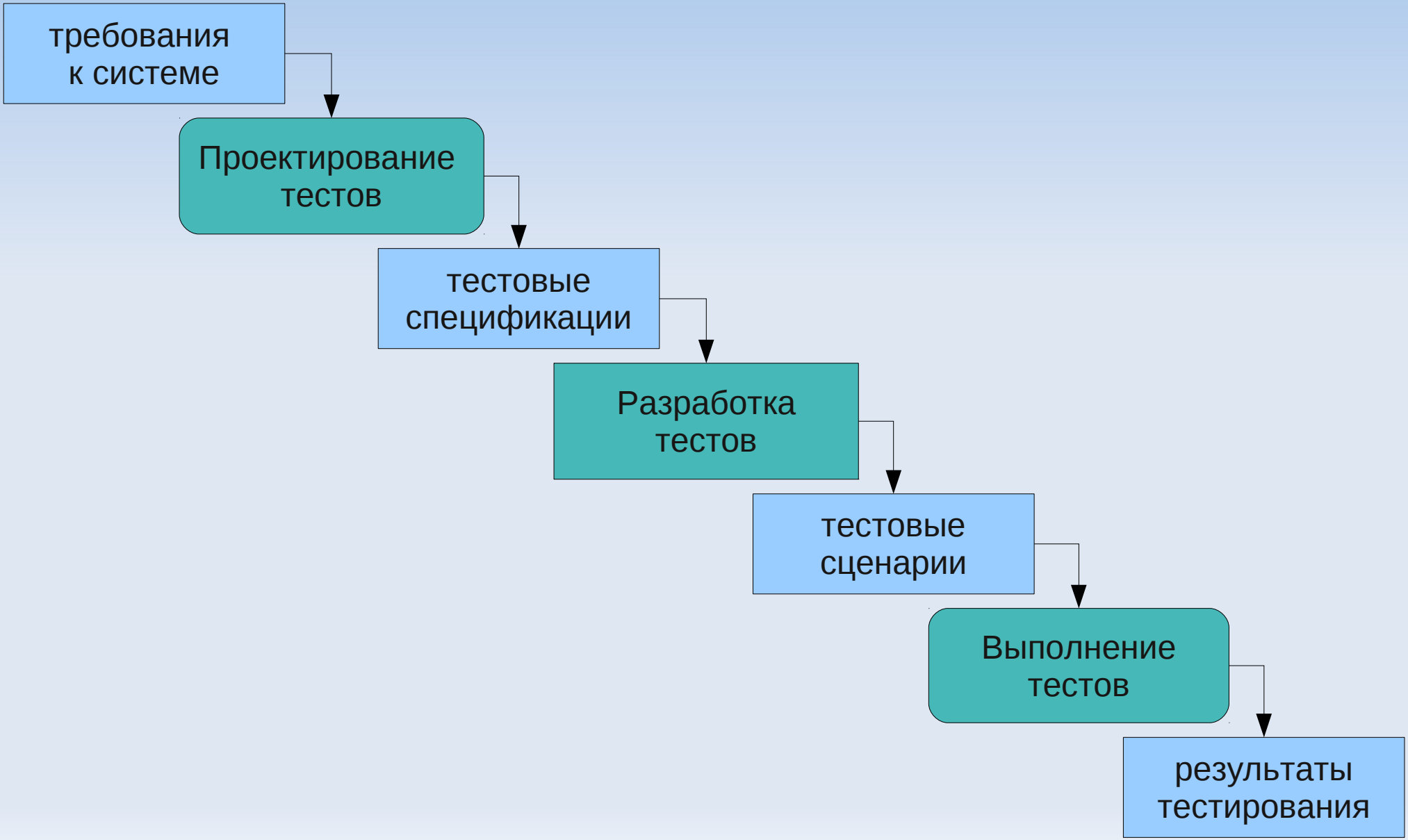
Системное тестирование (2)



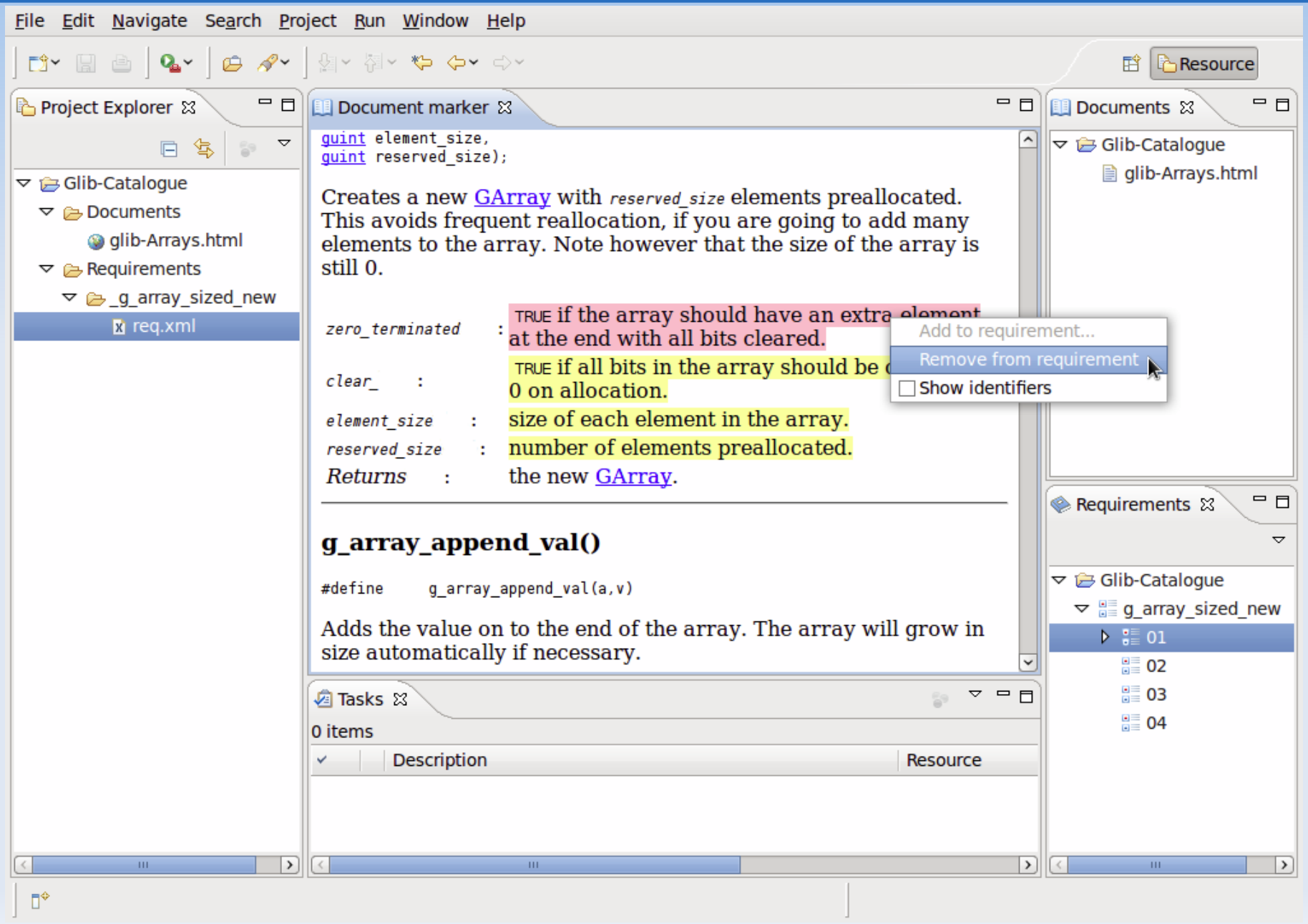
АРМ СТ (1)

- требования к системе
- Проектирование тестов
- тестовые спецификации
- Разработка тестов
- тестовые сценарии
- Выполнение тестов
- журналы тестирования

АРМ СТ (2)


требования к системе
Проектирование тестов
тестовые спецификации
Разработка тестов
тестовые сценарии
Выполнение тестов
результаты тестирования

APM CT (3)


File Edit Navigate Search Project Run Window Help
Resource
Project Explorer
Glib-Catalogue
Documents
glib-Arrays.html
Requirements
_g_array_sized_new
req.xml
Document marker
guint element_size,
guint reserved_size);
Creates a new GArray with reserved_size elements preallocated. This avoids frequent reallocation, if you are going to add many elements to the array. Note however that the size of the array is still 0.
zero_terminated : TRUE if the array should have an extra element at the end with all bits cleared.
clear_ : TRUE if all bits in the array should be 0 on allocation.
element_size : size of each element in the array.
reserved_size : number of elements preallocated.
Returns : the new GArray.
g_array_append_val()
#define g_array_append_val(a,v)
Adds the value on to the end of the array. The array will grow in size automatically if necessary.
Add to requirement...
Remove from requirement
Show identifiers
Documents
Glib-Catalogue
glib-Arrays.html
Requirements
Glib-Catalogue
g_array_sized_new
01
02
03
04
Tasks
0 items
Description
Resource

АРМ ПТ (4)

Ближайшие планы

- верификация AADL-модели архитектуры
- графический AADL-редактор
- развитие AFDX-симулятора
- реализация аналитических методов анализа AFDX-сети

Спасибо за внимание!

Хорошилов Алексей
khoroshilov@ispras.ru

ISP RAS