Восьмая независимая
научно-практическая конференция
«Разработка ПО 2012»

1 - 2 ноября, Москва

Две старые идеи в новой упаковке или старые методы решения новых задач

Петренко Александр Константинович

http://ispras.ru/~petrenko

ИСП РАН Институт Системного Программирования Российской Академии Наук

Случай из жизни

Аспирант прислал письмо с 15-ю вопросами.

Суть вопросов:

- Что такое наука?
- Чем ученые отличаются от инженеров?
- Чем наука отличается от обычной работы?
- и далее…

Соображение, которое появилось после двухлетних размышлений:

- В годы моей молодости таких вопросов не задавали…

Об истории двух научных идей

- Программирование «сверху-вниз»
  - Задача создания надежной авионики
- Сосредоточенное описание рассредоточенных действий
  - Задача обеспечения выполнения правил «безопасного программирования» (safety rules)

Идея № 1. Программирование «сверху-вниз»

Отцы-основатели:

- Top-down design was promoted in the 1970s by IBM researcher Harlan Mills and Niklaus Wirth. Mills developed structured programming concepts for practical use and tested them in a 1969 project to automate the *New York Times* morgue index. ..Niklaus Wirth, the developer of Pascal programming language, wrote the influential paper *Program Development by Stepwise Refinement*.

Harlan Mills

Программирование «сверху-вниз»

- можно вести при помощи разных методов и подходов. Общее в них то, что мы сначала принимаем некоторые общие решения, а затем их уточняем или детализируем. Это может касаться

  построения структуры системы (проектирование от более крупных подсистем к составляющим этих подсистем);

  разделения алгоритма на части (пролог, действие, эпилог)

  определение слоев/стека протоколов и так далее

Задача Д. Ван Тассела (см. «Стиль, разработка, эффективность, отладка и испытание программ»)

Одеть мужчину.

Первый уровень уточнения мог бы выглядеть так:

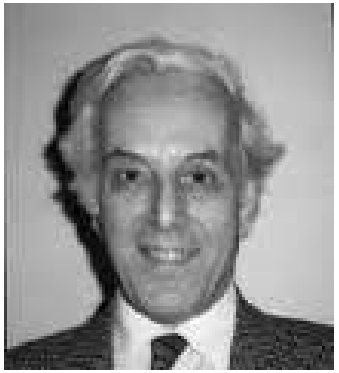
Одеть нижнюю половину
Одеть верхнюю половину

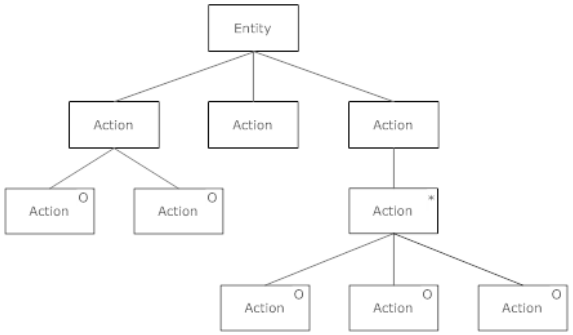
Задача Д. Ван Тассела (см. «Стиль, разработка, эффективность, отладка и испытание программ») (2)

Окончательный проект выглядит так:

Надеть трусы
Надеть брюки
Надеть ботинки
Надеть носки
Надеть майку
Надеть рубашку

JSP, JSD, SADT, SDL

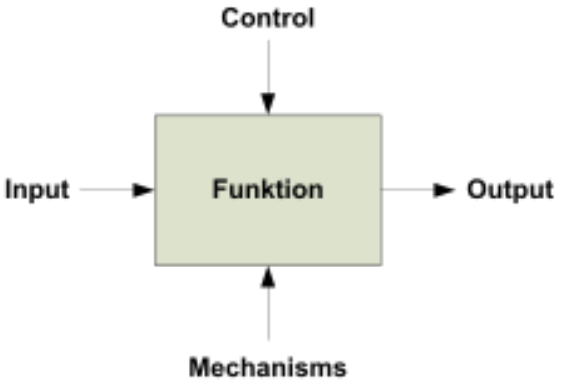


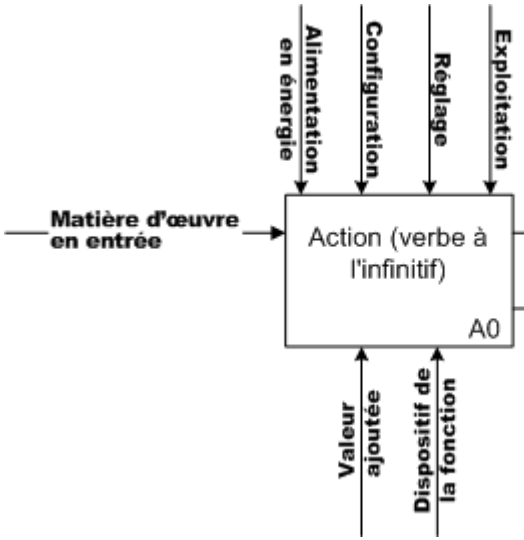


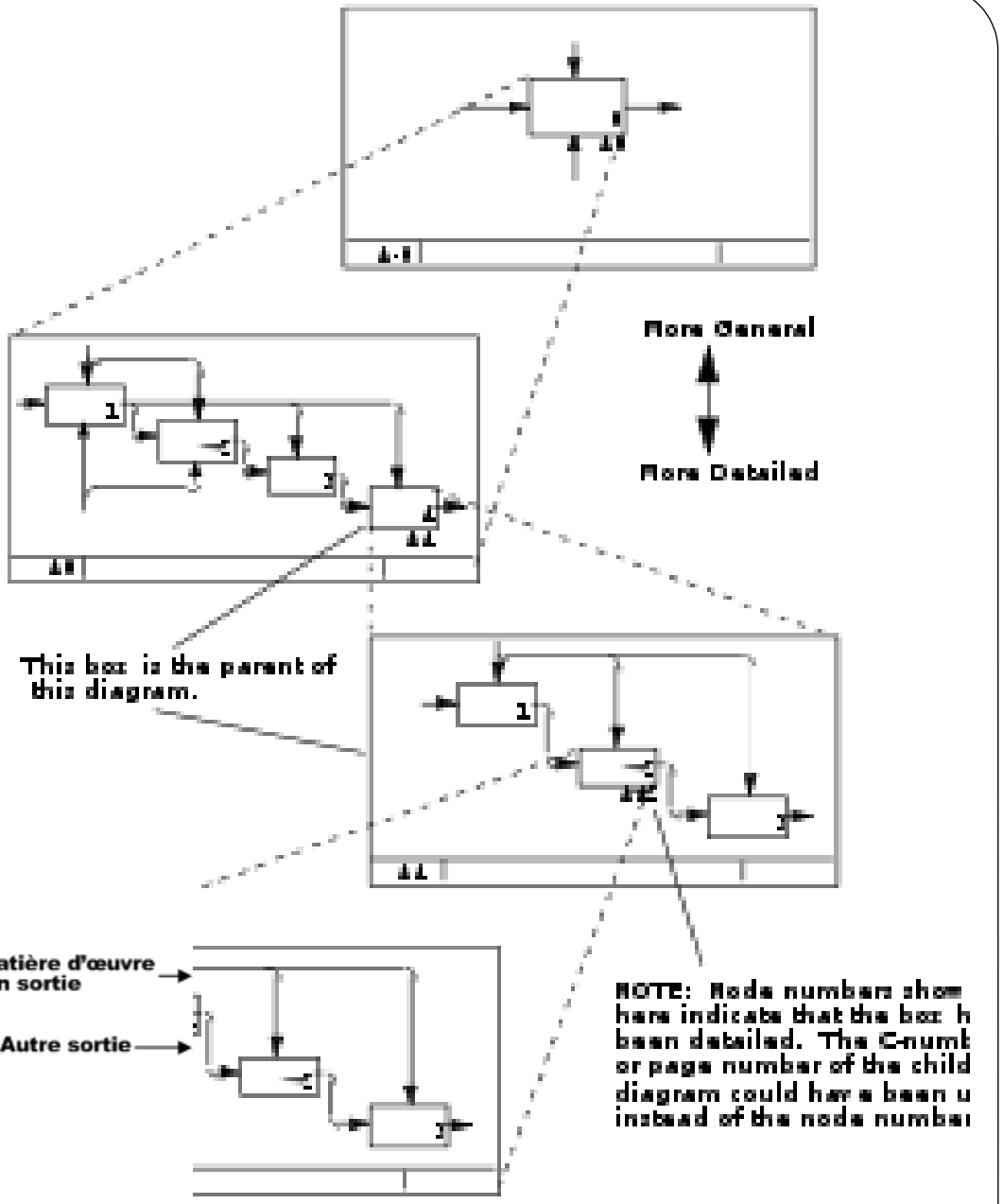
Майкл Джексон

- Jackson Structured Programming (JSP) – начало 1970-х
- Jackson System Development (JSD)
- **Structured Analysis and Design Technique** (SADT)
  - SADT 1969 to 1973 by Douglas T. Ross and SofTech, Inc. SADT was used in the MIT. It received extensive use starting in 1973 by the US Air Force.
- **Specification and Description Language** (SDL) — это язык спецификаций, предназначенный для получения описаний поведения реактивных и распределенных систем (1976).

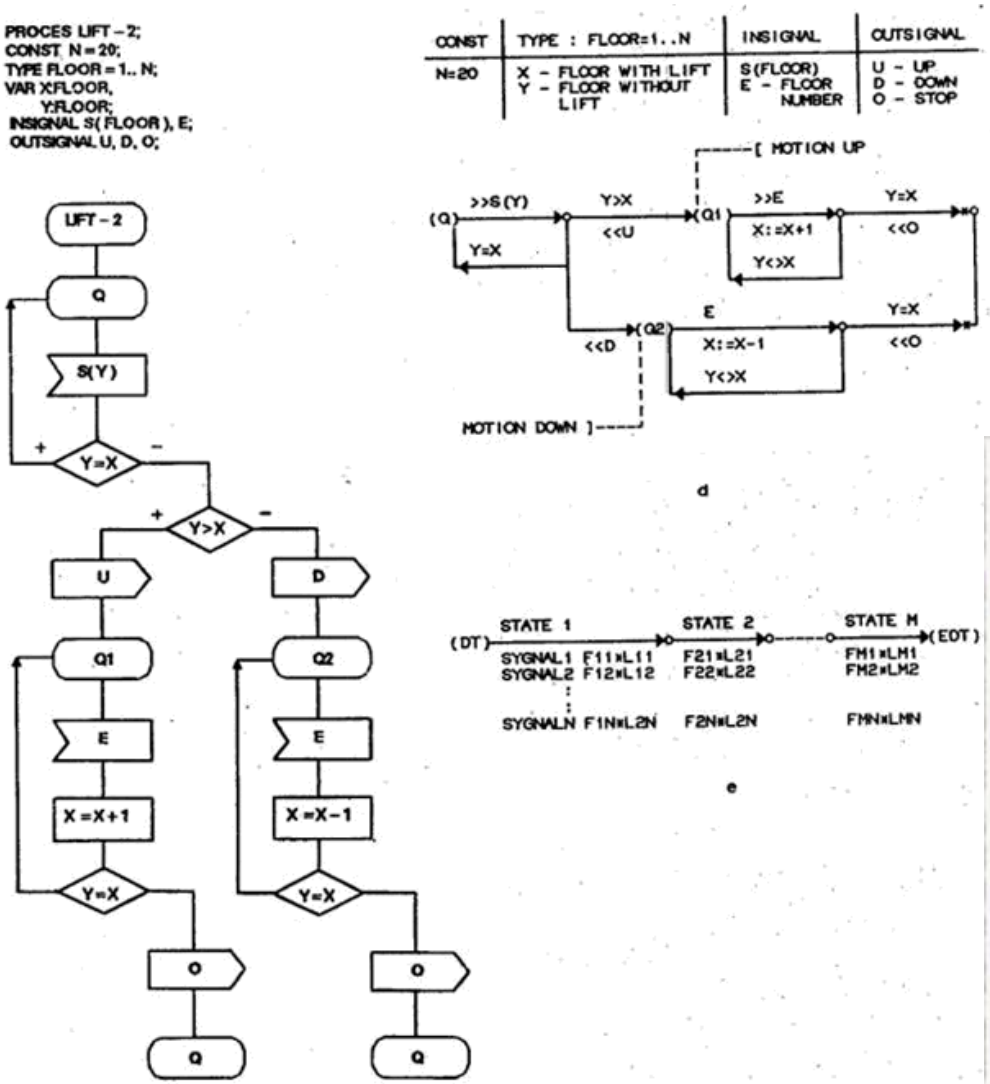
SADT


Control
Input
Funktion
Output
Mechanisms


Alimentation en énergie
Configuration
Réglage
Exploitation
Matière d'œuvre en entrée
Action (verbe à l'infinitif)
A0
Matière d'œuvre en sortie
Autre sortie
Valeur ajoutée
Dispositif de la fonction

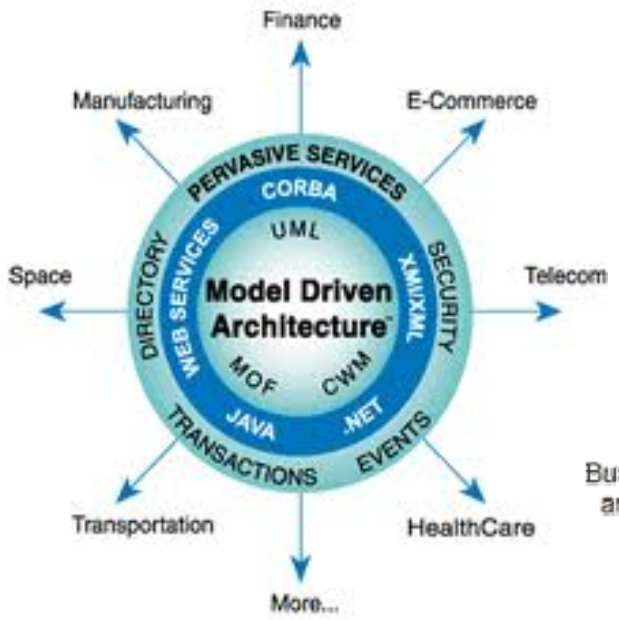

More General
More Detailed
This box is the parent of this diagram.
NOTE: Node numbers shown here indicate that the box h been detailed. The C-numb or page number of the child diagram could have been u instead of the node number

Советский ответ: Р-Технология

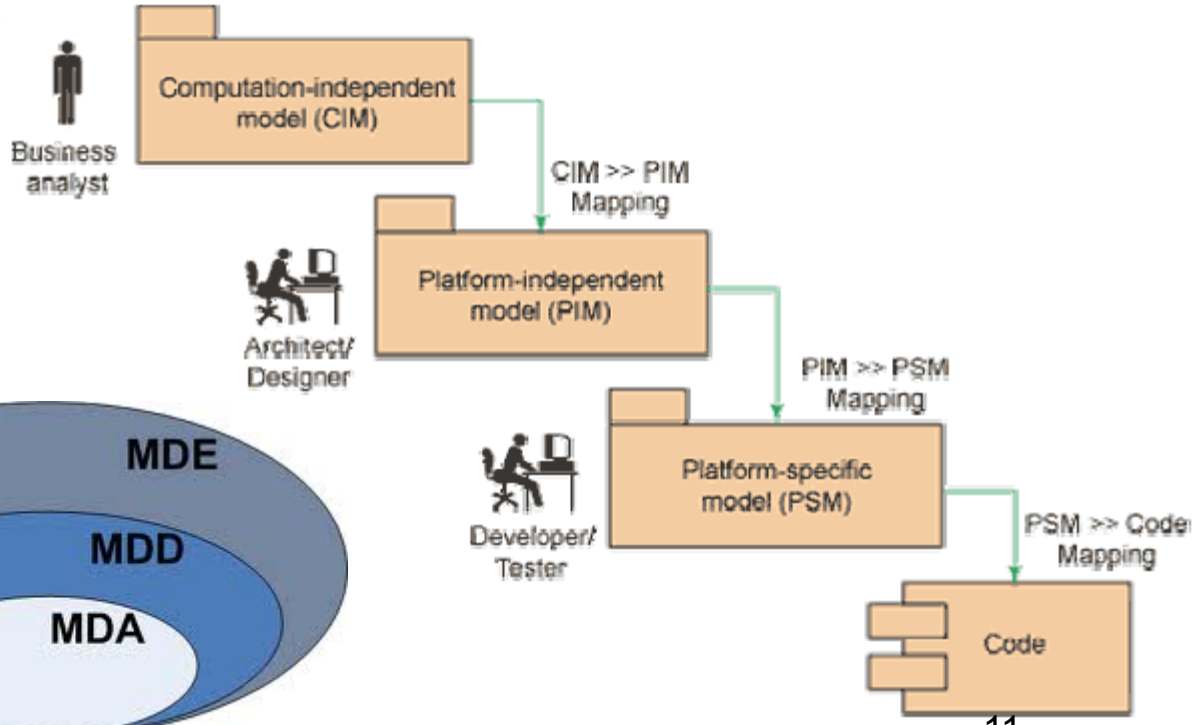


- И.В.Вельбицкий с академиком В.М.Глушковым

Крупнейшие современные инициативы



- Model Driven Architecture (MDA) - OMG



Перспективы

Новый импульс – ответственные системы. Гражданская авионика

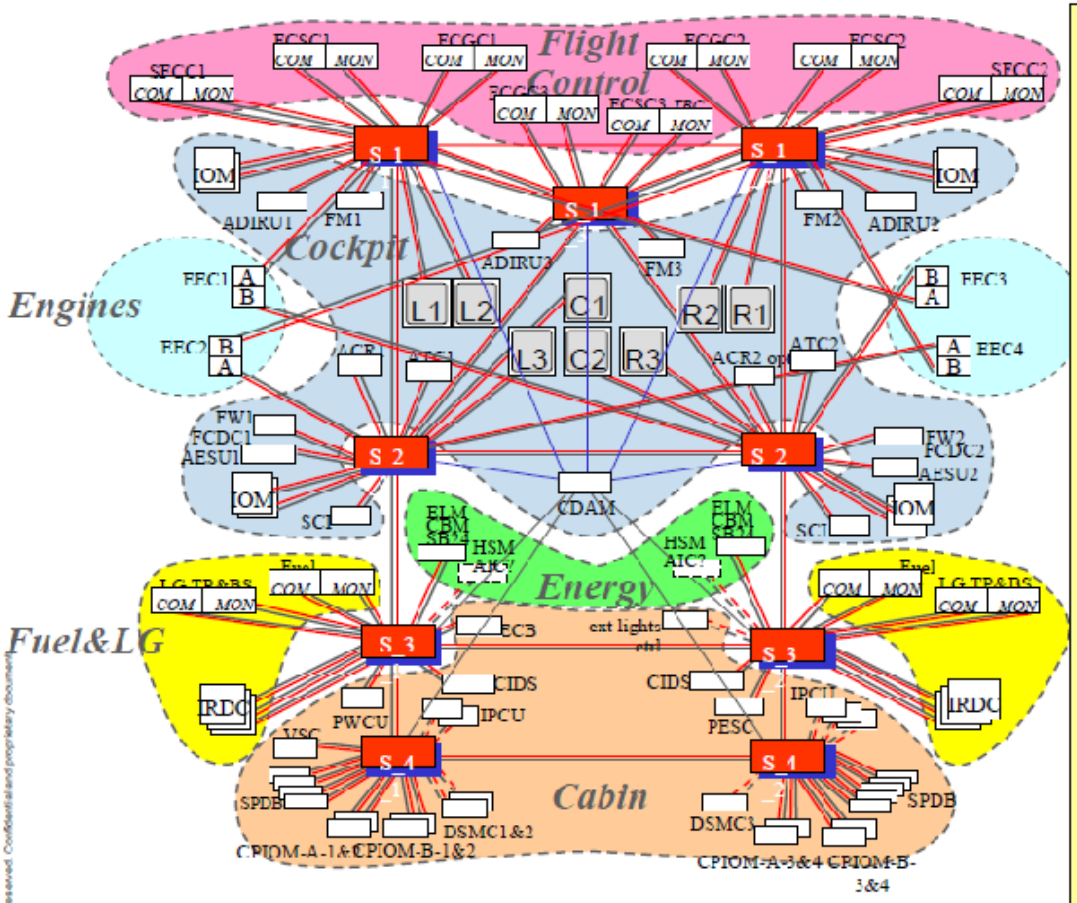
Академик Е.А.Федосов

- Программа развития Интегрированной Модульной Авионики - ИМА.

Авионика

- Более 100км проводов на одном самолете
- Несколько тонн веса
- Энергопотребление

ИМА на Airbus-380



AFDX Network:

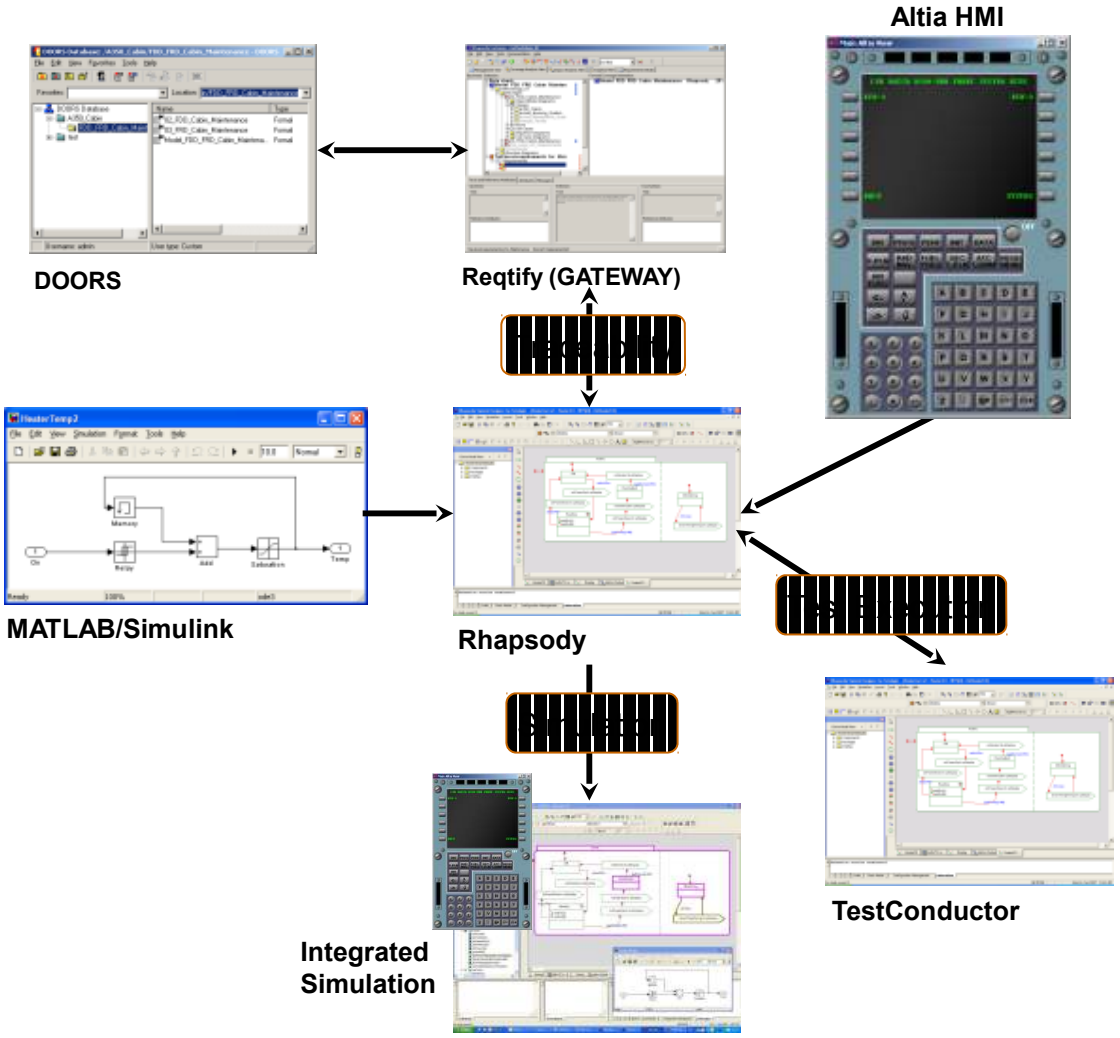
- 100 Mbits
- Redundant Network (A&B) with independent alimentation
- AFDX switches = 2 x 8
- NB of ports (connections) possible on each switch (20-24)
- MTBF of the switch is very high (100 000 hours expected)
- Up 80 AFDX subscriber

Новый ракурс разработки «сверху-вниз»

- Процессы разработки и сертификации идут параллельно
- Сертификация
  - Верификация
  - Отслеживание требований
  - Опережающая разработка и валидация требований

- Проектируется, не программа, а система.
  Новые объекты рассмотрения, новые средства, новые языки моделирования:
  - Ресурсы
  - Риски

Современный инструментарий разработки авионики

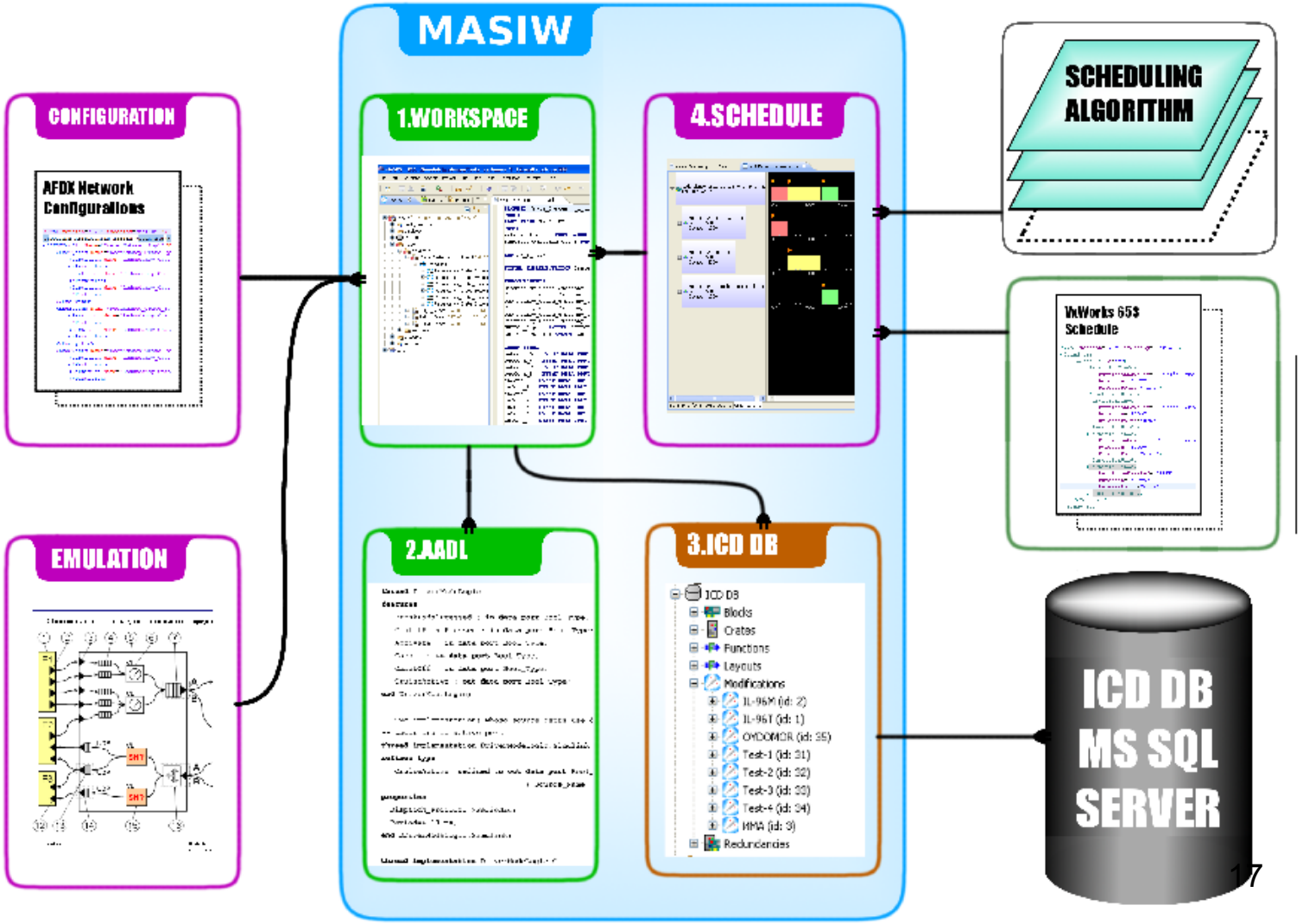


Tools:

- **DOORS (IBM/Telelogic)**
  - Requirements Management
- **Reqtify (TNI)**
  - Requirements Traceability
- **Rhapsody (IBM/Telelogic)**
  - Functional Behaviour
  - Interface Definition
  - Model Integration
  - Test definition
  - Simulation Execution
- **TestConductor (IBM/Telelogic)**
  - (Automatic) test execution
  - Coverage report generation
- **MATLAB/Simulink (MathWorks)**
  - Physical Behaviour
  - Environment Simulation
- **Altia HMI (Altia)**
  - Human Machine Interface Design
  - Simulation GUI

Инструментальный комплекс системного моделирования на **AADL**

(Architecture Analysis & Design Language)



Идея №2 или
Вечные вопросы программной инженерии

- Как правильно организовать эволюционное развитие программной системы
- Как правильно разбить на модули?

Первые ответы:

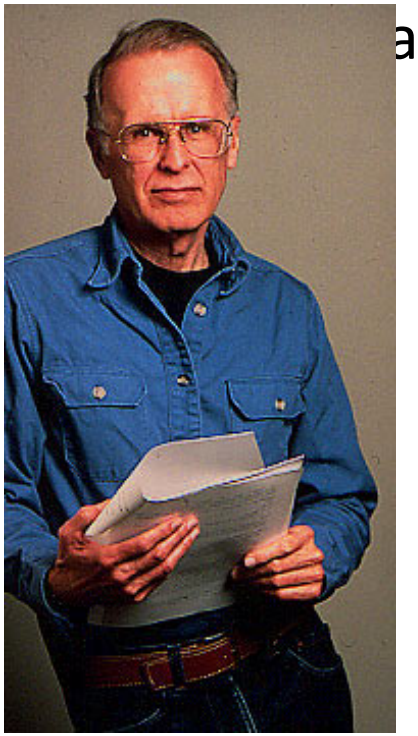
- Основной принцип – разделяй и властвуй, создавай подпрогра подпрограмм.

Морис Уилкс

Модули и языки программирования

- Джон Бэкус

Фортран (1957),
Алгол, BNF



а-Бура ИС-2

(1958)

Автор «Адресної мови»
першої мови програмування високого рівня
Ющонко Катерина Логвинівна

Каковы критерии правильного разбиения программы на модули?

Варианты:

- Модуль занимает не более 1 страницы
- В модуле не более 50 строк
- Модуль помещается в оперативной памяти
- …

К чему надо стремиться?

- Хорошая программа должна иметь низкое **сцепление** и высокую **связность**.

Каковы критерии правильного разбиения программы на модули?

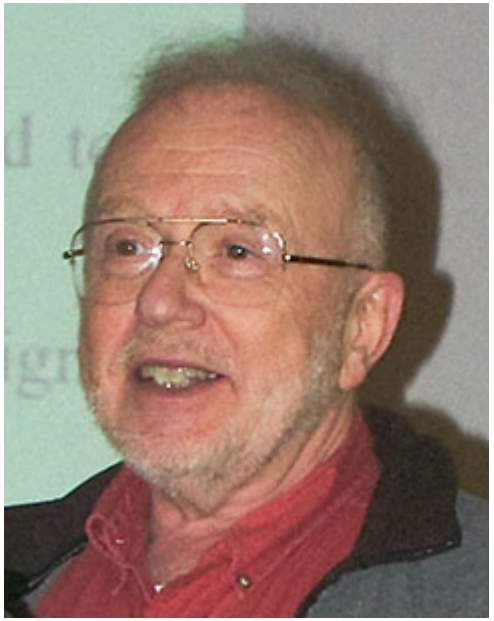
К чему надо стремиться?

- Хорошая программа должна иметь низкое **сцепление** и высокую **связность**

Связность модуля (Cohesion) — это мера зависимости его частей. Связность — **внутренняя** характеристика модуля. Чем выше связность модуля, чем «черней» его ящик (капсула, защитная оболочка модуля), тем меньше «ручек управления» на нем находится и тем проще эти «ручки».

Сцепление (Coupling) — мера взаимозависимости модулей поданным. Сцепление — внешняя характеристика модуля.

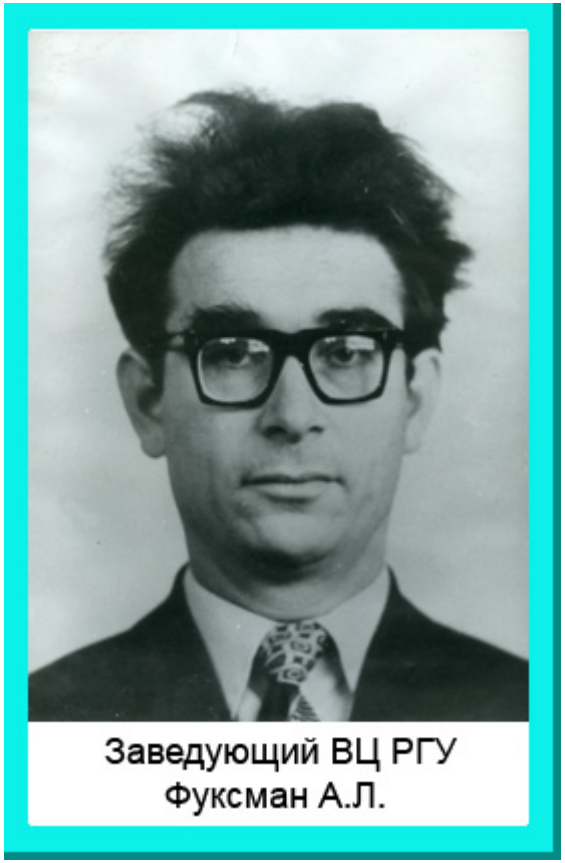
Дейв Парнас



- Первопричиной любого изменения является пересмотр какого-либо из решений, выбранных при формировании прежней версии программы. Поэтому реализацию каждого решения, принимаемого в ходе разработки,… надо оформлять в виде модуля. Тем самым последствия решения скрываются от остальных частей программы.

Цитируется по книге
М.М.Горбунова-Посадова

Какая технология должна поддерживать такую модульность?



Заведующий ВЦ РГУ
Фуксман А.Л.

Адольф Львович Фуксман

- …суть модульного программирования –
- сосредоточенное описание рассредоточенных действий.
- В отличие от Парнаса Фуксман предложил технологию слоистой разработки программ с соответствующими инструментальными средствами.

Новая парадигма и новые языки программирования – Аспектно-ориентированный подход

- **Gregor Kiczales** is a professor of computer science at the University of British Columbia in Canada. His best known work is on aspect-oriented programming and the AspectJ extension for Java
- **Владимир Олегович Сафонов** Профессор кафедры информатики мат-мех. Факультета СПбГУ

Новая задача – найти и проанализировать все потенциально небезопасные места в программе

- Пример подхода: найти и специфическим образом инструментировать фрагменты вида

```
spin_lock(lock);
…
…
buf = kmalloc(size, GFP_KERNEL);
…
…
spin_unlock(lock);
```

Для этого нужно описать некоторый дополнительный «аспект»

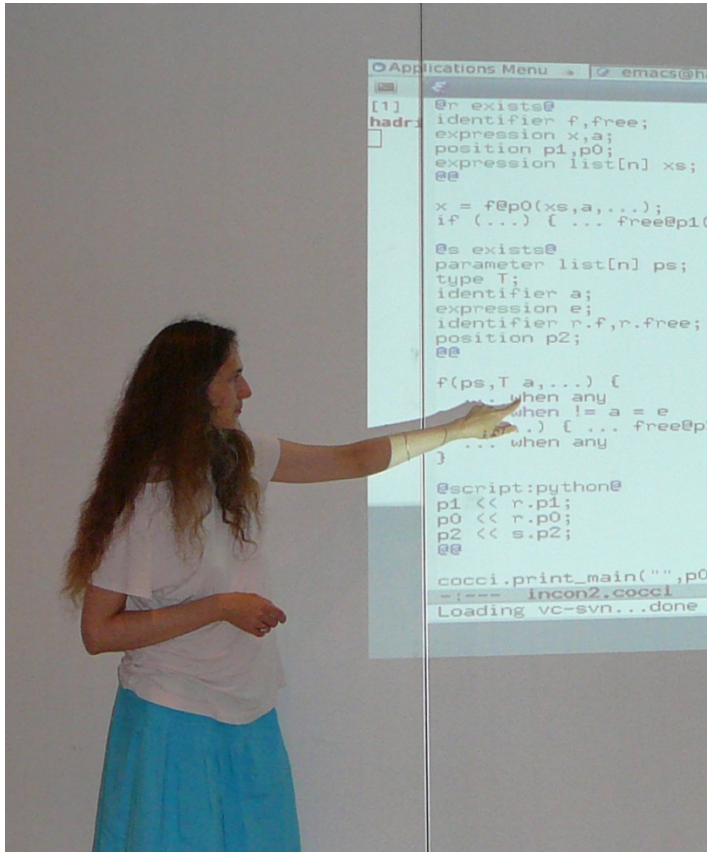
```
around: call(static inline void spin_lock(..)) {
  model_spin_lock();
}
around: call(static inline void spin_unlock(..)) {
  model_spin_unlock();
}

before: call(static inline void *kmalloc(.., gfp_t flags))
{
  check_flags(flags);
}
```

Результат инструментации

```
...
static inline void *aux_kmalloc(size_t size, gfp_t flags) {
  check_flags(flags);
  return kmalloc(size, flags);
}
...
int spinlock_held = 0;
void model_spin_lock() {
  spinlock_held = 1;
}
void model_spin_unlock() {
  spinlock_held = 0;
}
void check_flags(gfp_t flags) {
  if (spinlock_held && flags != GFP_ATOMIC) {
    ERROR: goto ERROR;
  }
}
```

Практические применения



- Технология SLAM – верификация драйверов ОС Windows
- Технология LDV (Linux driver verification)
- Coccinelle - Инструмент поддержки эволюции ядра OS Linux

Юлия Лаволь *(Julia Lawall), INRIA*

«Виден ли свет в конце тоннеля?»

или,

- Какие дороги (туннели) мы выбираем (О’Генри)

«Виден ли свет в конце тоннеля?»

«Виден ли свет в конце тоннеля?»

- **Мексиканские археологи недавно обнаружили под знаменитым храмом Змей в древнем городе Теотиукан тоннель, который закупорили предположительно 1800 лет назад.**
- Он имеет вид лабиринта длиной 120 метров на глубине 15 метров.

- **Ученые**

«Виден ли свет в конце тоннеля?»

Спасибо!

Что могу порекомендовать:

MBT/ETAPS Workshop: http://mbt-workshop.org
Spring Young Researcher's Colloquium (SYRCoSE): http://syrcose.ispras.ru
Seminar on Software Design&Analysis Technologies (SDAT): http://sdat.ispras.ru