

# Генерация моделей программно-аппаратных комплексов в соответствии с заданными ограничениями

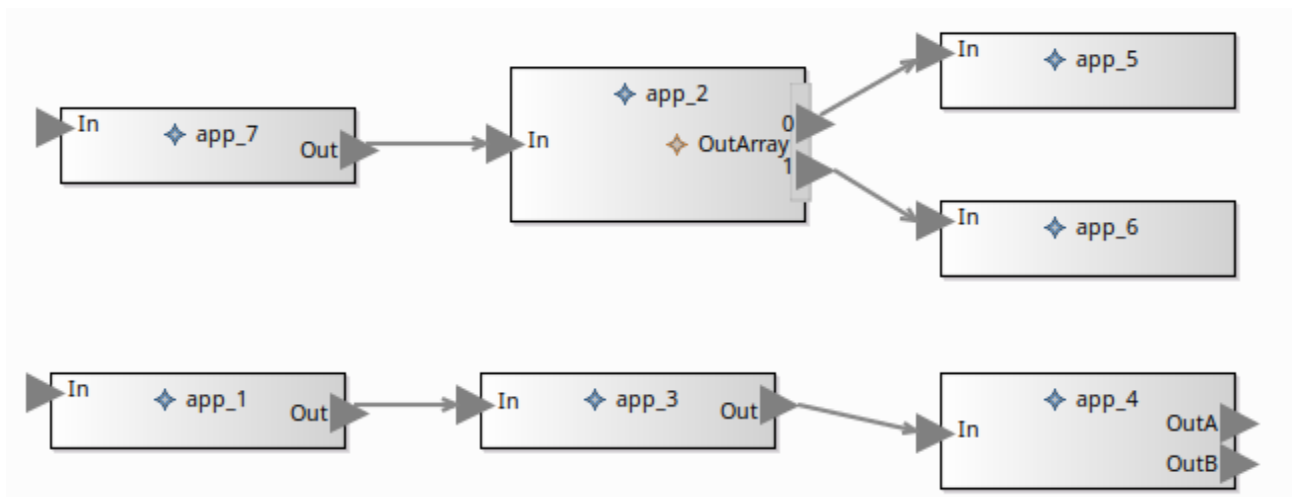
## 1 Входные данные

Пояснение: далее в тексте будет использоваться термин 'Ресурс'. Это именованное свойство элемента программно-аппаратного комплекса, которая характеризуется физической величиной. Тот или иной элемент комплекса может либо предоставлять 'Ресурс', либо потреблять. Набор 'Ресурсов' (имена, предоставляемое или потребляемое количество) задаётся для каждого элемента индивидуально. Элемент может как потреблять, так и предоставлять 'Ресурсы' (включая один и тот же ресурс).

### 1.1 Программная составляющая

Набор функциональных приложений, взаимодействующих друг с другом. Функциональное приложение – это программа (процесс), которая выполняется на вычислительной машине. Для выполнения своей работы функциональному приложению требуются ресурсы: вычислительное время процессора, память (ОЗУ, ПЗУ), и т. п. Функциональные приложения взаимодействуют друг с другом через порты ввода/вывода. Для обеспечения обмена информацией между приложениями так же требуются вычислительные ресурсы, только сетевые.

Пример (картинка):



Пример описания приложения:

Функциональное приложение 'app\_1' требует следующие ресурсы:

'RAM' = 4 МВ (требуется ресурс с именем 'RAM' в количестве 4 мегабайта)

'ROM' = 1 МВ (требуется ресурс с именем 'ROM' в количестве 1 мегабайта)

'CPU' = 20000 Оп / 1 s (требуется ресурс с именем 'CPU' в количестве 20000 операций в секунду)

Что касается ресурса 'CPU' – представлено некорректное физическое измерение данного ресурса. Этот вопрос необходимо будет проработать.

Связь 'app\_1.Out' → 'app\_3.In' требует следующие ресурсы:

'BANDWIDTH' = 1 KB / 1 s (от сетевого интерфейса вычислительного модуля и шины передачи данных требуется пропускная способность величиной 1 килобит в секунду).

## 1.2 Аппаратная составляющая

Набор типов вычислительных машин, сетевого оборудования и оборудования энергоснабжения. Каждый тип аппаратного компонента характеризуется набором интерфейсов для взаимодействия с другими аппаратными компонентами и вычислительными ресурсами.

Пример описания типа вычислительного модуля:

Модуль 'CPU\_type\_1' предоставляет следующие ресурсы:

'RAM' = 1 GB

'ROM' = 1 TB

'CPU' = 1 TOP / 1 s (вычислительный модуль способен выполнять 1000 гигаопераций в секунду, в таком подходе не учитывается возможность наличия нескольких процессоров/ядер в модуле и много чего другого, которое оказывает влияние на скорость выполнения того или иного кода, но хорошего решения пока не найдено)

потребляет следующие ресурсы:

'POWER\_U' = 20 V (для работы модуля необходимо напряжение 20 V)

'POWER\_A' = 1 A (для работы модуля необходим ток 1 A)

имеет следующие сетевые интерфейсы:

'eth\_0', который предоставляет следующие ресурсы:

'BANDWIDTH' = 100 Mb / 1 s

'eth\_1', который предоставляет следующие ресурсы:

'BANDWIDTH' = 1 Gb / 1 s

Имеет следующие интерфейсы для подключения источников энергии;

'power\_in', который потребляет и одновременно предоставляет следующие ресурсы (интерфейс сможет предоставить ресурсы для модуля, если необходимые ресурсы будут предоставлены этому интерфейсу):

'POWER\_U' = 20 V

'POWER\_A' = 1 A

## 1.3 Ограничения

Все программные компоненты не должны потреблять больше ресурсов, чем может предоставить аппаратный компонент, за которым закреплён программный. В общем случае в полученной итоговой модели программно-аппаратного комплекса потребители ресурсов должны быть связаны с поставщиками ресурсов. При чём потребителем не обязательно будет только программный компонент. Например, вычислительный модуль является потребителем энергии, которую поставляет источник энергии (оба компонента аппаратные).

Кроме этого должны быть учтены дополнительные ограничения. Которые задаются на основе набора типов дополнительных ограничений.

Набор типов дополнительных ограничений с одной стороны является фиксированным в момент запуска процесса генерации модели программно-аппаратного комплекса (и должен быть зафиксирован в рамках данной работы), но с другой стороны должна быть возможность добавления новых типов ограничений без переработки основной реализации алгоритма генерации модели.

Примеры ограничений:

Ограничения программных компонентов:

1. два заданных функциональных приложения не должны выполняться на одном вычислительном модуле;
2. обмен данными по двум заданным соединениям между функциональными приложениями не должен производиться через одни и те же сетевые компоненты;
3. два заданных функциональных приложения должны обязательно выполняться на одном вычислительном модуле.

Ограничения аппаратных компонентов:

1. два вычислительных модуля, за которыми закреплены функциональные приложения с ограничением №1 из 'Ограничения программных компонентов' не должны получать энергию от одного источника энергии;
2. сетевые компоненты, которые используются для передачи данных через соединения между функциональными приложениями с ограничением №2 из 'Ограничения программных компонентов' не должны получать энергию от одного источника энергии.

## 2 Задача

Разработать программу на языке Java.

Программа в качестве входных данных получает:

1. описание модели программной составляющей;
2. набор типов доступных аппаратных компонентов;
3. набор дополнительных ограничений.

Для заданных входных данных программа должна сгенерировать модель аппаратной составляющей (аппаратная составляющая генерируется путём создания необходимого количества экземпляров аппаратных компонентов на основе заданного набора типов компонентов) и создать общую модель программно-аппаратного комплекса путём связывания элементов программной составляющей с аппаратной (в общем случае, путём закрепления потребителей ресурсов с поставщиками).

При этом полученная модель программно-аппаратного комплекса должна удовлетворять всем ограничениям (как ресурсным – не должно быть ни одного потребителя ресурса, не связанного с поставщиком соответствующего ресурса, так и заданными дополнительно).

В рамках данной работы потребуются решить следующие связанные задачи:

1. определить способ описания моделей: элементов, связей между элементами, ресурсов, связей между поставщиками и потребителями ресурсов и т. п.; то, что приведено в разделе 'Входные данные' можно (но не обязательно) использовать как базис для более детального описания способа декларации моделей;
2. решить проблему с ресурсом, определяющим вычислительную мощность вычислительного модуля (задача очень сложная, поэтому можно остановиться на упрощённом варианте типа предложенного ранее в данном документе);
3. предложить базовый набор ресурсов (имена и единицы измерения), который рекомендуется использовать при описании входных данных; но это не означает, что программа должна поддерживать только этот набор ресурсов, в общем случае какие ресурсы (как ресурсы будут называться и какой физической величиной будут описываться) должен решать пользователь программы;

4. предложить базовый набор дополнительных ограничений; при решении этой задачи необходимо учитывать следующее:
  - программно-аппаратные комплексы должны обладать определённой надёжностью, один из способов – резервирование функциональности; в программной модели ответственные функциональные приложения и связи между ними будут представлены в нескольких экземплярах для резервирования друг друга и, соответственно, сбой в одном из аппаратных компонентов не должен приводить одновременно к потере работоспособности резервирующих друг друга программных компонентов; поэтому нужны ограничения, которые позволят сгенерировать такую модель;
  - время отклика комплекса на какое-либо событие может играть важную роль, в таких случаях важную роль начинает играть время передачи информации от одного приложения к другому; при чём информация может передаваться не только напрямую от приложения к приложению, а и через приложения посредники (которые обеспечивают определённую обработку данных); соответственно, нужны ограничения, которые позволят задать время доставки сообщений.

### 3 Рекомендации

Для работы с моделями желательно использовать Eclipse Modeling Framework (<https://www.eclipse.org/modeling/emf/>).

Для работы с физическими величинами можно использовать JSR-385 (<https://www.jcp.org/en/jsr/detail?id=385>), реализация <https://github.com/unitsofmeasurement>.