

Лабораторная работа 3.

Целью лабораторной работы является знакомство студентов с методологией IDEF0 для проведения декомпозиции сложной задачи на подзадачи.

Студенты должны получить и закрепить на практике следующие знания и умения:

1. Использовать методологию IDEF0 для функционального моделирования при решении поставленной задачи.
2. Выполнять декомпозицию сложной задачи на подзадачи.
3. Знать и уметь применять на практике требования стандарта IDEF0 при построении диаграмм, визуализирующих результаты декомпозиции.
4. Использовать программное приложение Ramus Educational в качестве инструмента для построения IDEF0-диаграмм.

1. Общее задание

1. Выполнить декомпозицию нижеприведенной задачи на подзадачи.
2. Результат декомпозиции представить в формате IDEF0-диаграмм, используя приложение Ramus Educational. Диаграммы оформить в соответствии с требованиями стандарта IDEF0.
3. Файлы *.rsf (результат работы Ramus Educational) поместить в репозиторий в папку Lab_03.
4. В Wiki создать отдельную страницу, на которой разместить эти же решения, но уже как png-файлы (пункт меню «Диаграмма»/«Экспортировать как рисунки» в Ramus Educational).

2. Условие задачи

Исправить ошибку в программе по ее указанному описанию.

Входными данными являются адрес репозитория Git и номер ошибки, указанный в Issue в GitLab, выходными – измененный репозиторий и закрытая ошибка.

Требования к решению задачи:

1. Количество уровней декомпозиции должно приводить к однозначному решению задачи.
2. Обязательно должны быть приведены команды Git, которые использовались для решения поставленной задачи.

3. Методология функционального моделирования. Описание стандарта IDEF0

IDEF0 - методология функционального моделирования, которая предусматривает визуальное представление изучаемой и/или разрабатываемой системы в виде набора взаимосвязанных функций (функциональных блоков). При этом для визуализации используется специальный графический язык.

Основные понятия стандарта IDEF0:

- функциональный блок (обозначает некоторую конкретную функцию в рамках рассматриваемой системы либо с использованием глагола, либо с использованием отглагольного существительного);

- интерфейсная дуга (отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображенную данным функциональным блоком);
- декомпозиция (разбиение сложного процесса на составляющие его функции);
- глоссарий (описание сущности данного элемента посредством соответствующих определений, ключевых слов, повествовательных изложений и т.д.).

Каждая из четырех сторон функционального блока имеет своё определенное значение (роль), при этом:

- левая сторона имеет значение «Вход» (детали, сырье, финансы, документы, информация и данные и т.п.);
- правая сторона имеет значение «Выход» (в контексте результата работы функции);
- верхняя сторона имеет значение «Управление» (инструкции, методические рекомендации, математические и/или физические методы и/или законы и пр.);
- нижняя сторона имеет значение «Механизм» (сотрудники, станки, компьютеры и программное обеспечение и пр.).



Моделирование в рамках IDEF0 всегда начинается с разработки *контекстной диаграммы*, на которой система представляется как единое целое. Контекстная диаграмма содержит только один функциональный блок и обозначается идентификатором «А-0».

В процессе декомпозиции, функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации, для чего используется диаграмма второго уровня, которая содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы и называется дочерней (Child diagram) по отношению к нему.

Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока.

Важно отметить, что в каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме. Однако часто возникают случаи, когда отдельные интерфейсные дуги не имеет смысла продолжать рассматривать в дочерних диаграммах ниже какого-то определенного уровня в иерархии, или наоборот – отдельные дуги не имеют практического смысла выше какого-то уровня. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования. Однако в рамках курса «Основы программной инженерии» данный механизм использоваться не будет.

Принципы ограничения сложности IDEF0-диаграмм

- количество функциональных блоков на диаграмме должно быть от 3 до 6. Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание;
- количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг целесообразно ограничить четырьмя.

7. Литература

1. <https://habrahabr.ru/company/trinion/blog/322832/>
2. <http://www.interface.ru/home.asp?artId=3443>
3. Презентация к лекции №3.