

# АВТОМАТИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ СИСТЕМЫ ИНФОРМАЦИОННОЙ ЗАЩИТЫ ПРЕДПРИЯТИЯ СРЕДСТВАМИ IDEF И UML

А. Ю. АТИСКОВ, Т. В. МОНАХОВА

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<panthernator@yandex.ru>

---

УДК 681.3

Атисков А. Ю., Монахова Т. В. Автоматизация процесса проектирования системы информационной защиты предприятия средствами IDEF и UML // Труды СПИИРАН. Вып. 3, т. 2. — СПб.: Наука, 2006.

**Аннотация.** Приведена технология построения функциональной модели системы защиты в IDEF и дальнейшей автоматизированной трансформации в UML. Также рассмотрен переход от построенной модели к программному коду. — Библ. 3 назв.

UDC 681.3

Atiskov A. J., Monahova T. V. Automation of designing the system of informational protection of enterprise using IDEF and UML technologies // SPIIRAS Proceedings. Issue 3, vol. 2. — SPb.: Nauka, 2006.

**Abstract.** Technology of designing functional diagrams of protection system with IDEF and semi-automatic transformation them into UML diagrams of classes are mentioned. Conversion from model to a framework of code is shown. — Bibl. 3 items.

---

## 1. Постановка задачи

Современная инструментальная база проектирования средств защиты информации опирается на отдельные методологии, например, Allfusion Process Modeler (IDEF0, IDEF3, DFD) или Rational Rose и Together (UML). Не существует инструментов, поддерживающих взаимосвязь моделей, построенных с использованием данных типов нотаций. Поэтому предлагается технология проектирования, включающая трансформацию функционального описания системы защиты в объектно-ориентированное. Данный подход позволяет использовать возможности UML в области проведения реинжиниринга программного кода при включении новых компонент в систему (например, блока идентификации по отпечатку пальцев). Более того, изменения в диаграммах классов UML можно сразу же перенести на программный код за счет автоматической генерации на различных языках программирования. С другой стороны, мы не теряем возможности наглядно проектировать бизнес-логику функционирования новых блоков в IDEF0-диаграммах.

В [1] функциональное моделирование системы информационной безопасности было рассмотрено на примере информационной системы военного-медицинского учреждения. На рис. 1 приведен фрагмент IDEF0-диаграммы функциональной схемы защиты информации для данного примера. Как видно из рисунка, защита информации имеет следующие основные подфункции: контроль над входом в систему и выполнением задач, а также мониторинг. Эти процессы тесно связаны между собой, и каждый из них также имеет ряд подпроцессов.

В данном случае функциональное проектирование позволяет описать систему с помощью общепринятых в России стандартов. Стоит заметить, что есть

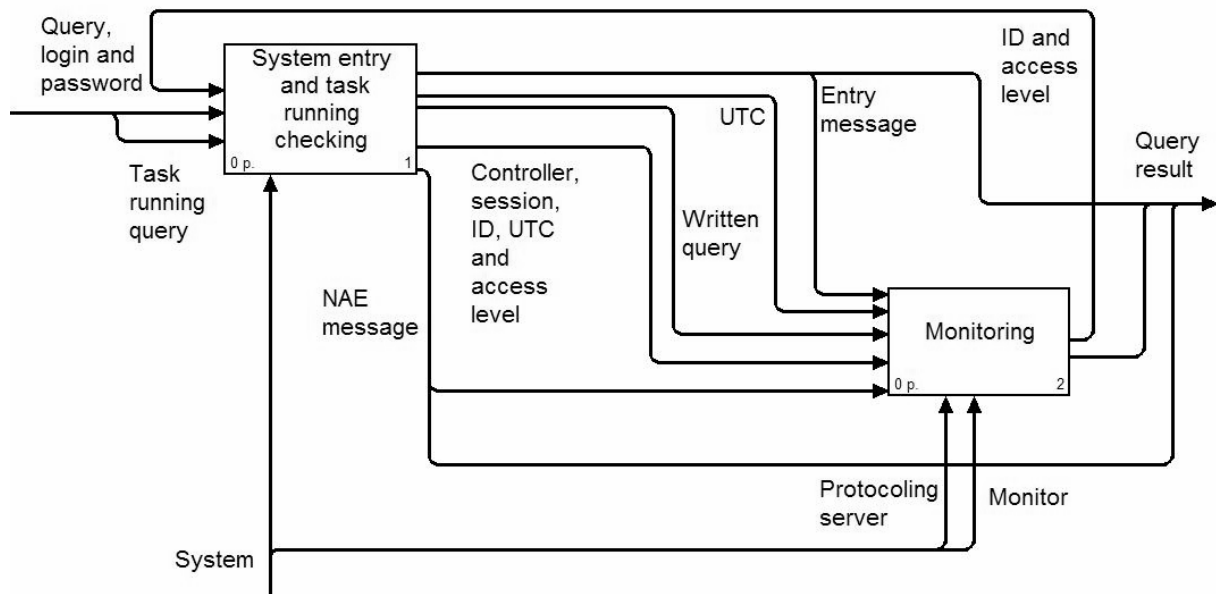


Рис. 1. Фрагмент диаграммы защиты информации в нотации IDEF0.

возможность сочетать различные виды IDEF при построении одной функциональной модели, хотя в нашем примере мы использовали только IDEF0.

Рассмотрим фрагмент диаграммы, приведенный на рис. 1, более подробно. Для этого приведем ниже диаграмму дерева узлов для данной части модели. Как сказано в [2], дерево узлов представляет собой иерархию функций, выполняемых моделируемой системой, а также позволяет рассмотреть модель целиком, но без указания входных и выходных данных и т.п.

Как показано на диаграмме дерева узлов, приведенной на рис. 2, функция контроля над входом в систему и выполнением задач имеет следующие подфункции: получение имени и пароля, запрос на авторизацию; авторизация; получение списка задач, ID, УВК и уровня доступа, создание сессии; создание формы пользовательского интерфейса. В свою очередь, мониторинг декомпозируется на наблюдение за входом в систему и наблюдение за запуском задач, которые также состоят из подфункций.

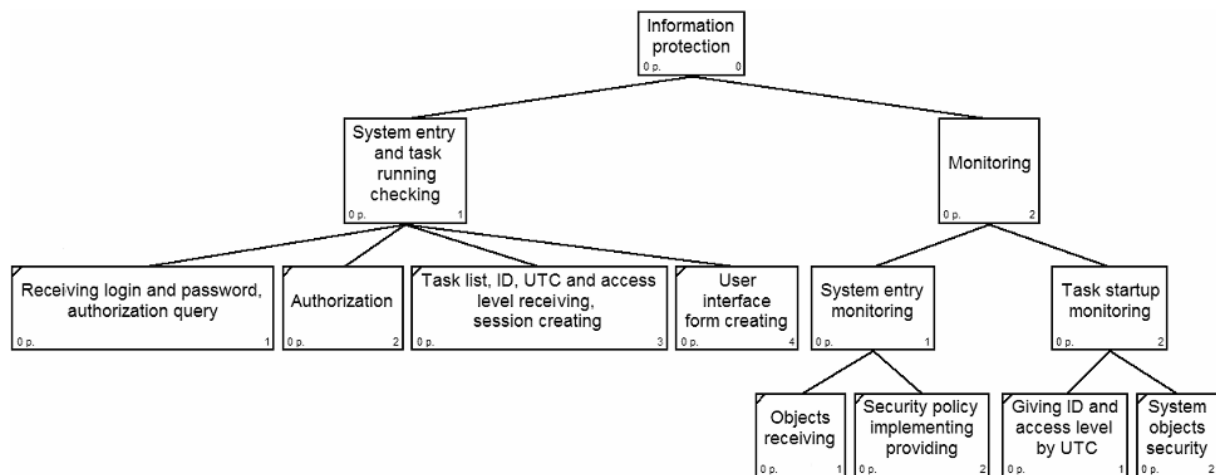


Рис. 2. Диаграмма дерева узлов для фрагмента модели.

## 2. Проведение трансформации функциональных диаграмм в диаграммы классов

К сожалению, по приведенным выше диаграммам невозможно напрямую получать программный или хотя бы каркас будущей программной системы.

Однако существуют методы проектирования, которые позволяют на основе построенных в них моделей, получать шаблон приложения. Среди этих методов особенно успешным и проработанным является методология UML, содержащая в своем составе возможность моделирования диаграмм классов с последующей генерацией кода по ним. Существует ряд CASE-средств, поддерживающих UML — например Together и Rational Rose, но они не работают с IDEF-диаграммами. Процесс представления диаграмм бизнес-процессов в виде модели программного обеспечения является трудоемким процессом. Сложность обусловлена применением технологически разрозненных методов и средств описания бизнес-процессов. Не все классы и объекты, определенные в бизнес-архитектуре могут быть включены в модель программного обеспечения. Поэтому трансформация бизнес-модели в модель программного обеспечения не является строго-формализованным процессом [3].

В ходе исследований моделирования системы информационной защиты вообще и перехода от IDEF к UML в частности, была разработана программная система, позволяющая трансформировать IDEF-диаграммы в диаграммы языка UML. Применяя систему трансформации, решаем задачу перевода структурной модели непосредственно в каркас кода.

Особенность программной реализации состоит в том, что имеется возможность работы с моделями на любой стадии разработки (независимо от количества блоков на диаграмме), использовать данные о названиях и роли стрелок и названиях блоков, а также проводить автоматическую трансформацию. При этом можно открыть полученную модель классов непосредственно в окне Together, не используя промежуточные программные средства.

Это единственная на данный момент система, позволяющая осуществить такое преобразование, которое является основой для перехода от описания предметной области и организации будущей информационной системы к ее программному коду.

После трансформации программой диаграмм IDEF0 была получена диаграмма классов (рис. 3, сделан из среды проектирования Together).

По данной диаграмме видно, как распределяются функции по всей системе, и какие объекты должны будут отвечать за их непосредственное выполнение. В качестве примера можно рассмотреть класс `Monitor` — он обладает методами `Giving_ID_and_access_level_by.UTC` (с возвращаемым значением класса `ID_and_access_level`) и `System_entry_monitoring` (с возвращаемым значением класса `Protocolling_server`), при этом этот класс связан с классами `Entry_message` и `Written_query` и другими.

Полученная диаграмма сразу показывает степень сложности классов и наиболее сильные связи между ними, что позволяет правильно оценить сложность дальнейшей реализации системы. Данная диаграмма могла быть получена и ручным способом, путем создания классов и связей в среде Together, но при этом это заняло гораздо больше времени и могло сопровождаться синтаксическими ошибками. Также можно заметить, что класс `Monitor` напрямую не связан с классом `System`, что может свидетельствовать о том, что их разработ-

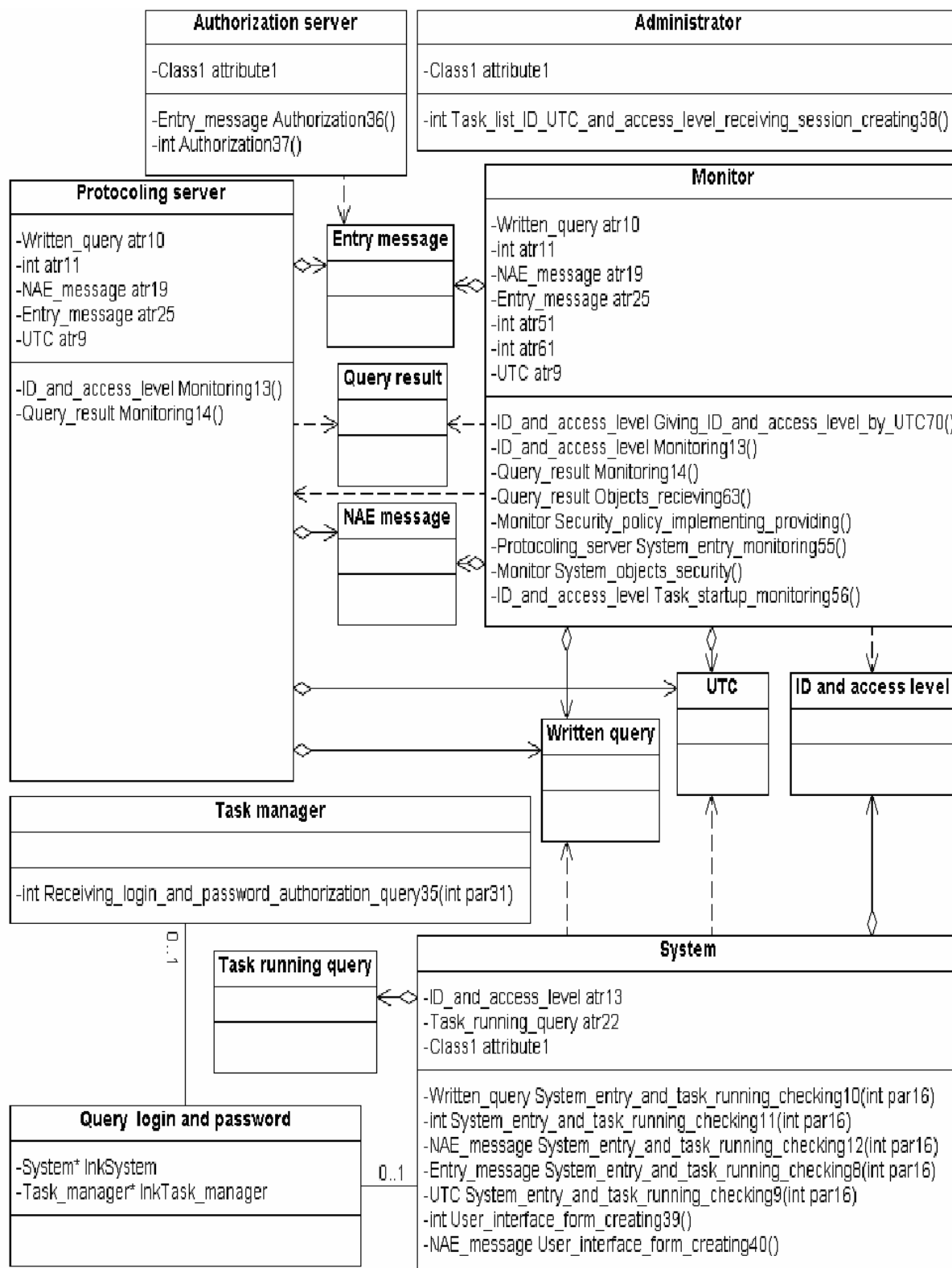


Рис. 3. Фрагмент UML-диаграммы классов, полученной с помощью указанной системы.

ку можно поручить независимым разработчикам. Стоит отметить, что эта диаграмма не является окончательной — при проведении цикла реинжиниринга возможно внесение новых атрибутов, методов и связей. Кроме того, из некоторых классов можно убрать дублирующиеся методы.

По данной диаграмме классов мы получили каркас кода. Ниже приводим пример кода для класса (сгенерированного автоматически Together):

```
#include "Entry_message.h"
class Authorization_server {
private:

    /**
     * @shapeType DependencyLink
     */
    /*# Class1 attribute1; */

    int Authorization37();
    Entry_message Authorization36();
}
```

### 3. Заключение

Таким образом, для получения программного кода проектируемой системы информационной защиты требуется провести предпроектный анализ, составить тезаурус предметной области, построить IDEF-диаграммы, перевести их в диаграммы UML и применить к ним одну из систем проектирования (Rational Rose, Together или другую), что существенно упрощает процесс моделирования по сравнению с отдельным составлением UML и IDEF-диаграмм. Это позволяет снизить трудоемкость этапа функционального моделирования информационной системы и, в конечном счете, ускорить весь процесс построения системы.

### Литература

1. *Воробьев В. И., Монахова Т. В.* Функциональное моделирование системы информационной защиты предприятия // Труды СПИИРАН. Вып. 2, т. 2. СПб.: Наука, 2005. С. 216–222.
2. *Маклаков С. В.* Моделирование бизнес-процессов с BPwin. М.: Диалог-МИФИ, 2002. 224 с.
3. *Атисков А. Ю., Воробьев В. И.* Автоматизированная система трансформации диаграмм бизнес-процессов в диаграммы классов // Труды СПИИРАН. Вып. 3, т. 2. СПб.: Наука, 2006. [В настоящем издании].