

УДК 681.3.06.14

**СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ КОНСТРУКТОРА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ОБРАБОТКИ СИГНАЛОВ**

Л.С. Файнзильберг, Т.Н. Жук, А.Г. Шаховал

Международный научно-учебный центр информационных технологий и систем НАН Украины и МОН Украины

e-mail: fainzilberg@voliacable.com

Компьютерная обработка сигналов сложной формы, например, электрокардиограмм (ЭКГ), магнитокардиограмм (МКГ), реограмм и других биологических сигналов, порождаемых живым организмом, часто требует разработки нетривиальных алгоритмов, которые не всегда удается получить формальными методами. Поэтому конструктор прикладной информационной технологии (ИТ) должен на модельных и реальных данных проводить многочисленные эксперименты по оценке эффективности того или иного алгоритма прежде чем принять окончательное решение о его включении в состав ИТ [1].

В докладе с позиций объектно-ориентированного подхода развивается концепция построения инструментальной системы, ориентированной на поддержку принятия таких решений.

Будем рассматривать структуру ИТ обработки сигнала как технологическую цепочку конечного числа N_Q вычислительных процедур (элементарных обработок), посредством которых вектор $Y = (y[1], \dots, y[K])$ дискретных значений наблюдаемого сигнала последовательно преобразуется в выходной продукт – оценку \hat{s} текущего состояния объекта:

$$D_R^{(1)} = A_1(\alpha_1, Y) \rightarrow \dots \rightarrow D_R^{(i)} = A_i(\alpha_i, D_S^{(i)}) \rightarrow \dots \rightarrow D_R^{(N_Q)} = A_{N_Q}(\alpha_{N_Q}, D_S^{(N_Q)}) = \hat{s}, \quad (1)$$

где $A_i(\cdot)$ – алгоритм преобразования исходных данных $D_S^{(i)}$ в результат $D_R^{(i)}$ обработки на i -й стадии, α_i – параметры настройки алгоритма, причем в общем случае под исходными данными i -й стадии понимается не только результат $D_R^{(i-1)}$, но и любые предшествующие ему результаты.

Если структура (1) определена и конструктор ИТ имеет полную информацию о каждой i -й стадии обработки $D_R^{(i)} = A_i(D_S^{(i)})$, то синтез ИТ сводится к традиционным этапам жизненного цикла разработки программного обеспечения. Однако опыт показывает, что при синтезе новых ИТ основное время затрачивается не столько на программную реализацию вычислительных алгоритмов, сколько на экспериментальные исследования, направленные на поиск подходящих алгоритмов и их оптимальную настройку.

Основная идея, которую предлагается положить в основу построения инструментальной системы состоит в том, что с ее помощью конструктор прикладной ИТ должен иметь возможность

- *проводить* необходимые эксперименты для выбора и оптимальной настройки вычислительных процедур, реализующих отдельные стадии обработки сигнала;
- *реализовать* технологическую цепочку алгоритмов обработки, используя готовые вычислительные компоненты самой инструментальной среды;
- *расширять* при необходимости состав компонентов инструментальной системы (рис. 1).

Анализ поставленной задачи с позиций объектно-ориентированного подхода позволил обосновать структуру двух основных программных классов, на основе которых

реализуется ядро инструментальной системы, удовлетворяющей сформулированным принципам.

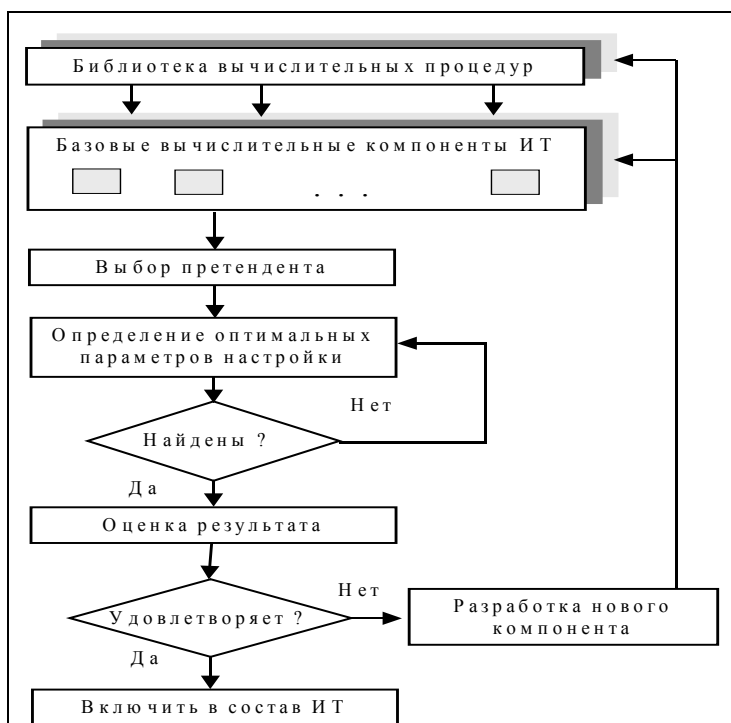


Рис. 1. Концептуальная идея построения системы

Первый класс представляет собой обобщенную модель носителя данных **МНД**, которая рассматривается в виде тройки

$$\text{МНД} = \langle D_C, D_G, G_M \rangle, \quad (2)$$

где D_C – поля содержательных данных, D_G – поля графических данных, G_M – методы отображения и модификации графических образов. Объекты, порождаемые на основе **МНД**, обеспечивают

- хранение содержательных данных, необходимых для организации элементарных вычислительных процедур на различных стадиях обработки сигнала;
- отображение результатов обработки отдельных стадий в виде экранных графических образов и текстовой информации;
- доступ к данным через их экранные графические образы;
- модификацию графических образов (перемещение, масштабирование, разметка, изменения типа отображения и т.п.) (рис. 2):

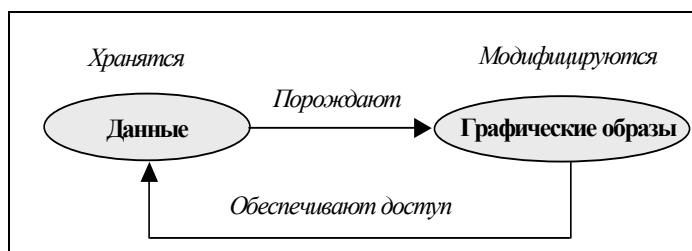


Рис. 2. Концептуальная идея построения МНД

Второй класс представляет собой обобщенную модель обработки данных (**МОД**). **МОД** представляет собой абстрактный класс с универсальными полями, свойствами и

методами, инвариантными относительно конкретной процедуры обработки. При этом **МОД** содержит виртуальный метод «пустой» обработки, который доопределяется в наследниках этого класса (рис. 3).

Абстрактный класс **МОД** позволяет унифицировать понятие элементарной обработки данных, упорядочить процесс создания программных компонент, реализующих отдельные стадии обработки, и тем самым организовать удобный интерфейс взаимодействия между активными экземплярами (объектами) класса **МНД** и конкретной процедурой обработки.

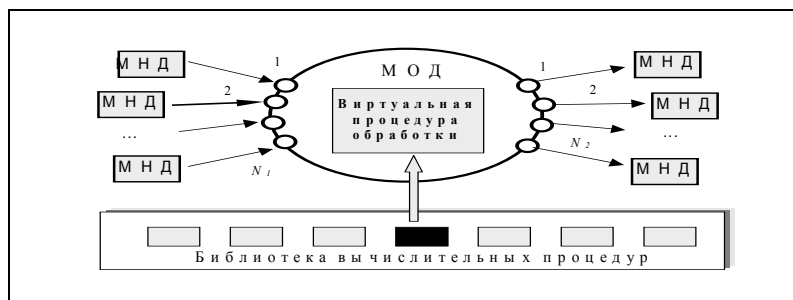


Рис. 3. Упрощенная схема компоновки элементарной стадии обработки:

МОД – модель обработки данных, **МНД** – модель носителя данных

МОД поддерживает унифицированный интерфейс вызова любой вычислительных процедуры:

$$\langle \text{Имя} \rangle (I_1, \dots, I_{N_1}; L_1^{(I)}, \dots, L_{N_1}^{(I)}; O_1, \dots, O_{N_1}; L_1^{(O)}, \dots, L_{N_2}^{(O)}; \alpha_1, \dots, \alpha_{N_{1\alpha}}; \text{Mess}), \quad (3)$$

где $I_1, \dots, I_{N_1}; O_1, \dots, O_{N_1}$ – указатели на массивы входных и выходных данных;

$L_1^{(I)}, \dots, L_{N_1}^{(I)}; L_1^{(O)}, \dots, L_{N_2}^{(O)}$ – число элементов в соответствующих массивах;

$\alpha_1, \dots, \alpha_{N_{1\alpha}}$ – настроечные параметры процедуры обработки;

Mess - строка сообщения об исключительной (аварийной) ситуации.

Такой интерфейс позволяет распараллелить работу программистов, участвующих в создании инструментальной системы. Прикладные программисты могут сосредоточиться на своей главной задаче – реализации эффективных вычислительных процедур обработки данных, не заботясь об организации диалога с пользователем и методах графического представления результатов, а системные программисты – рассматривать процедуру обработки как некий «черный ящик».

На базе производных классов **МОД** реализован широкий набор вычислительных компонент (более 100 модулей), обеспечивающих обработку сигналов различной физической природы. Для оценки эффективности этих компонент в системе обеспечена возможность «наложения» гармонических, случайных и импульсных помех на реальные и тестовые сигналы, а также имитации различных других искажений, с которыми приходится иметь дело при решении практических задач предварительной обработки сигналов.

Доступ к данным осуществляется по принципу «Point and Select»: для доступа к данным достаточно указать мышью на соответствующий график в рабочем окне, а для вызова требуемой вычислительной процедуры – выбрать соответствующий пункт меню. Тем самым конструктору ИТ предоставляется возможность в интерактивном режиме реализовать цепочки вычислительных процедур из готовых компонент самой инструментальной системы (рис. 4).

Удобной функцией системы является режим отображения, при котором пользователь имеет возможность в одном окне перемещать и масштабировать один график на фоне остальных (режим «интеллектуальный осциллограф»). Полезной находкой системы является также подход, обеспечивающий интерактивную настройку параметров вычислительных процедур с использованием идей когнитивной компьютерной графики.

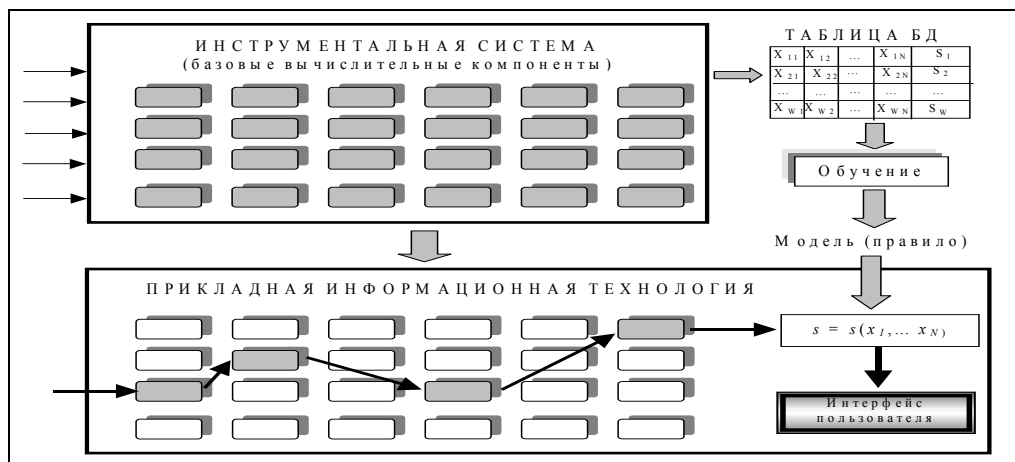


Рис. 4. Организация цепочки вычислительных процедур

На рис.5 показано рабочее окно системы при настройке процедуры сглаживания зашумленного сигнала (верхний график). Благодаря «виртуальной» связи органа управления и графического образа, отображающего результат обработки (нижний график), конструктор ИТ имеет возможность определить «оптимальное» значение параметра настройки вычислительной процедуры, визуально оценивая результат. Аналогично осуществляется настройка и других случаях, когда отсутствуют формальные методы определения оптимальных значений настроечных параметров.

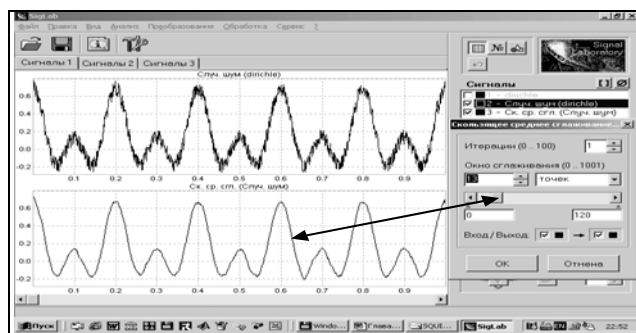


Рис. 5. Интерактивная настройка параметров вычислительного компонента

Инструментальная система позволила провести необходимые исследования при конструировании целого ряда прикладных систем обработки сигналов сложной формы, в частности, новых компьютерных технологий ФАЗАГРАФ и ИКАР для морфологического анализа ЭКГ и оценки variability сердечного ритма [2].

Литература

1. Файнзильберг Л.С. Синтез информационных технологий обработки сигналов в задачах косвенного контроля и диагностики // Управляющие системы и машины. – 1998. – № 2. – С. 40-47.
2. Fainzilberg L.S. Nowa metoda interpretacji zapisu EKG w balaniach skriningowych oraz w opiece domowej // Zdrowie publiczne.– 2005.– Vol. 115.– Number 4.– P.458-464.