Теоретическая модель процесса построения электронных форм для отображения и редактирования сложных структур данных

В. А. Скорняков

Рассматриваются вопросы математического описания информационных объектов и методов построения экранных форм для их отображения на экране и редактирования. Производится детальное изучение проблемы построения модели представления информационных объектов.

Введение

Модель содержания информационного объекта и модель формы представления этого объекта имеют различные предназначения, и вместе они предоставляют возможности создания и редактирования информационных объектов всех допустимых типов ([1]). Модели могут создаваться и редактироваться отдельно друг от друга и использоваться вместе при создании электронной формы, предназначенной для редактирования информационных объектов конкретного типа. Поэтому необходимо изучить вопрос соответствия моделей представления и моделей содержания ([1], [5]).

Рассмотрим процесс создания модели представления формы без учета необходимости однозначного отображения модели содержания на модель представления и обратно, а также возможность построения формы только по модели содержания, либо только по модели представления.

Для того чтобы содержание некоторого информационного объекта можно было отобразить или изменить с помощью формы, необходимо однозначное соответствие элементов модели представления формы элементам информационного объекта. В общем случае, это соответствие не является взаимно однозначным. Также, для того, чтобы с помощью некоторой формы можно было редактировать некоторый информационный объект, необходимо соответствие элементов объекта элементам этой формы. В общем случае, это соответствие не является взаимно однозначным.

Основные понятия теории

Узлы графа, представляющего собой модель представления электронной формы, принадлежат множеству W. Это множество счетно, т. к. состоит из конечного набора существующих элементов. Можно говорить, что W — пространство описателей элементов электронной формы. Множество всевозможных описателей, позволяющих наличие подчиненных вершин графа модели представления, обозначим как W_φ.

Определение 1

Узлы графа модели представления (см. [1]) электронной формы будем называть элементами формы.

Определение 2

Элементы электронной формы, созданные по любому из описаний множества W_φ называются контейнерами. Соответственно, элементы множества W_φ назовем контейнерами на множестве W.

Определение 3

Элементы электронной формы, отображающие вершины модели структуры типа A, называются повторяющимися контейнерами.

Определение 4

Элементы электронной формы, созданные по любому из описаний дополнения множества W_φ на пространстве W, назовем терминальными элементами формы. Соответственно, элементы множества $\overline{W_\varphi}$ назовем терминальными описателями элементов модели представления или терминальными элементами модели представления.

Определение 5

Семантическими блоками информационного объекта называются взаимосвязанные части этого объекта, на которые он разбивается [3].

Определение 6

Миниформа — это область формы, содержащая элементы формы, связанные единым алгоритмом расположения друг относительно друга в рамках миниформы (см. [1]).

Утверждение 1

Контейнер модели является семантическим блоком.

Доказательство

Способ выбора семантических блоков не оговаривается, а модель представления целиком разбивается на блоки из W_φ и связи между ними.

На основании утверждения 1 можно заключить, что семантическая модель документа изоморфна модели представления электронной формы в виде блоков из W_φ.

Рассмотрим пространство моделей представления. Это пространство содержит графы моделей представления, составленные из элементов множества W. Кроме того, по определению модели представления, любой такой граф в качестве корневого узла содержит миниформу. И обратно: любой ориентированный граф, состоящий из вершин множества W и не имеющий циклов, представляет собой модель представления электронной формы. Пространство моделей представления будем обозначать Ω.

В дальнейшем будем использовать некоторые обозначения из [1]:

$\varphi(c, v)$ — описание электронной формы, где c — модель содержания формы, v — модель представления формы.

Соответствие модели содержания и модели представления

Пусть M — пространство моделей структуры объектов, Ω — пространство моделей представления форм. Рассмотрим следующие преобразования: $\eta : M \to \Omega$, $\theta : \Omega \to M$. Используя эти преобразования, описание формы $\varphi(c, v)$ можно представить в виде функции одной переменной: $\varphi(c, \eta(c))$ или $\varphi(\theta(v), v)$.

Взаимно однозначное соответствие модели структуры объекта и модели представления

Если $\eta^{-1}(v) = \theta(v)$, то $\varphi(c, v) = \varphi(c, \eta(c)) = \varphi(\theta(v), v)$. В противном случае $\varphi(c, \eta(c)) \neq \varphi(\theta(v), v)$.

Пусть $\varphi = \varphi(c_1, v_1)$, где $c_1 = \theta(v_1) = \eta^{-1}(v_1)$, $v_1 = \eta(c_1) \Rightarrow \varphi(c_1, v_1) = \varphi(\theta(v_1), v_1) = \varphi(\eta^{-1}(v_1), v_1) = \varphi(\eta^{-1}(v_1), \eta(c_1)) = \varphi(c_1, \eta(c_1))$. То есть если есть взаимно однозначное преобразование $M \to \Omega$, то форму можно строить автоматически только по одной из моделей.

Для построения взаимно однозначного преобразования $M \to \Omega$ сделаем следующее:

1. Возьмем множество SAE = {S, A, E}, где S, A, E — типы вершин, из которых состоят элементы пространства M (см. [1]).
2. Рассмотрим пространство W типов элементов представления формы. Это пространство содержит всевозможные визуальные элементы

управления, позволяющие пользователю изменять как данные, так и структуру объектов из D. Множество W не является равномощным множеству SAE, следовательно, между ними невозможно установить взаимно однозначного соответствия. То есть для того, чтобы получить взаимно однозначное преобразование $M \to \Omega$, необходимо выделить из W всевозможные подмножества, равномощные SAE и среди этих подмножеств попытаться выбрать область определения взаимно однозначного преобразование $M \to \Omega$ таким образом, чтобы каждому из элементов SAE можно было поставить в соответствие единственный элемент указанного подмножества W. То есть нужно выбрать такое множество $W_m \subset W$, что для $\forall s \in SAE\ \exists! w \in W: \eta(s) = \eta^{-1}(w)$. Очевидно, что множество W_m состоит из трех элементов: $W_m = \{S_\eta, A_\eta, E_\eta\}$.

3. Для того чтобы сохранялась обратимость ребер графов из M, элементы S_η, A_η, E_η будем выбирать таким образом, чтобы S_η был контейнером, позволяющим включать в себя элементы множества W_m, а A_η — был контейнером, позволяющим включать элементы типа S_η.

Утверждение 2

Существует взаимно однозначное отображение $\eta_e : M \to \Omega$.

Доказательство

$\forall m \in M$ представим в виде множества упорядоченных пар, которые соответствуют направленным ребрам графа m: $M_m = \{\{m_{a1}, m_{b1}\}, \{m_{a2}, m_{b2}\}, \ldots, \{m_{an}, m_{bk}\}\}$. Рассмотрим множество Ω_m, соответствующее M_m в виде множества пар элементов множества W_m, соответствующих ребрам графа пространства Ω: $\Omega_m = \{\{W_{ma1}, W_{mb1}\}, \{W_{ma2}, W_{mb2}\}, \ldots, \{W_{man}, W_{mbk}\}\}$. Это соответствие является взаимно однозначным, т. к. множества SAE и W_m взаимно однозначно отображаются друг на друга, а направленное ребро на множестве SAE однозначно отображается на направленное ребро на множестве W_m. В качестве существования элементов W_m, которые могут быть упорядочены операцией включения, выберем следующее множество: W_m = {миниформа, миниформа, текстовое поле ввода}.

В дальнейшем обозначение η_e будем использовать для обозначения взаимно однозначного отображения $M \to \Omega$.

Отображение модели структуры на модель представления

В различных практических задачах один объект d удобнее редактировать с помощью различных моделей представления. Например, отделу кадров важнее поля с адресом и квалификацией сотрудника, а отделу социального обслуживания интереснее информация о составе семьи и доходах того же сотрудника. Также часто требуется скрыть от всех пользователей часть данных. Поэтому рассмотрим более общее соответствие $\eta : M \to \Omega$, которое, в общем случае, не является взаимно однозначным. Однозначность теряется в тот момент, когда при преобразовании из модели представления удаляются элементы, соответствующие некоторым элементам модели структуры или элементы управления ставятся на разные позиции.

Областью определения η является пространство M, а областью значений η является пространство Ω. Пусть $m \in M$, тогда $\eta(m) = \Omega_m \in \Omega$. То есть Ω_m — множество моделей представления, которым соответствует модель некоторой структуры m.

Так как модель представления создается для удобства пользователя, то не имеет смысла использовать для редактирования объект, для типа структуры которого не может существовать связи хотя бы с одним элементом модели представления. В этом случае пользователь будет видеть несколько полей формы, которые ничего не значат или, что еще хуже, данные, введенные в эти поля, будут теряться при последующей обработке данных, введенных пользователем. Поэтому в качестве множества Ω_m значений преобразования $\eta(m)$, будем рассматривать такие модели представления, элементы которых возможно связать с различными элементами модели структуры m. На практике соответствие η может реализовывать какой-либо автомат или пользователь, создающий модель представления объекта по имеющейся модели структуры этого объекта.

Напомню, что по определению модели представления, каждый элемент модели представления может быть связан с единственным элементом модели структуры объекта. Очевидно, что в этом случае для модели структуры m невозможно найти модель представления, содержащую элементов больше, чем $\omega = \eta_e(m)$. В качестве вырожденного случая модели представления можно рассмотреть миниформу, не содержащую в себе никаких элементов. В этом случае с учетом условий, описанных выше, на экране можно будет отобразить некоторый объект, но при этом не будут видны его данные и пользователь никак не сможет изменить какие-либо данные этого объекта.

Определение 6

Пусть $m \in M$, $\omega \in \Omega$ и $\omega = \eta_e(m)$. В этом случае модель представления ω будем называть максимальной на множестве $\Omega_m \subset \Omega$: $\eta(m) = \Omega_m$ и обозначать ω_{mmax}.

Определение 7

Пусть $m \in M$, $\omega \in \Omega$ и ω — модель представления, состоящая из пустой миниформы. В этом случае модель представления ω будем называть минимальной на множестве $\Omega_m \subset \Omega$: $\eta(m) = \Omega_m$. и обозначать ω_{min}.

Очевидно, что остальные элементы множества Ω_m получаются из ω_{mmax} удалением некоторых его элементов.

Утверждение 3

Объект любого типа можно отобразить с помощью модели представлений ω_{min}.

Доказательство

Для $\forall m \in M$ и $\Omega_m \subset \Omega$: $\eta(m) = \Omega_m \Rightarrow \omega_{min} \in \Omega_m$.

Утверждение 4

Пусть $m_1 \in M$. Тогда $\exists m_2 \in M$, $m_2 \neq m_1$: $\eta(m_1) \cap \eta(m_2) \neq \varnothing$.

Иными словами, можно найти такие модели структуры, что объекты, задаваемых ими типов, возможно отобразить и редактировать с помощью одних и тех же моделей представления.

Доказательство утверждения 2 следует из существования взаимно однозначного отображения η_e: $M \to \Omega$.

Рассмотрим класс множеств W, порожденный преобразованием η.

Теорема 1

Класс множеств $W \subset \Omega$, порожденный всевозможными соотношениями η: $M \to \Omega$, представляет собой алгебру пространства Ω.

Доказательство

Пусть $m \in M$ и $\eta(m) = \Omega_m \subset \Omega \Rightarrow \Omega_m \subset W$. Докажем, что $\overline{\Omega_m} \subset W$. Это верно, т. к. для $\forall \omega \in \overline{\Omega_m}$ $\exists m_\omega \in M$: $\eta_e(m_\omega) \in \overline{\Omega_m}$ $(m_\omega = \eta_e^{-1}(\omega))$. Следовательно, $\overline{\Omega_m} \subset W$.

Теперь докажем, что для $\forall \Omega_1, \Omega_2 \subset W$ $\Omega_1 \cup \Omega_2 \subset W$. То есть для $\forall \Omega_1, \Omega_2 \subset W$, $\Omega_3 = \Omega_1 \cup \Omega_2$, $\exists m_3 \in M$, η_3: $\eta_3(m_3) = \Omega_3 \subset W$. Возьмем модели структуры m_1 и m_2 и вместо корневых узлов этих графов создадим один новый элемент типа S. Полученный граф принадлежит множеству **D** (см. [1]), т. к. удовлетворяет всем необходимым условиям для графов этого множества. Получили некоторую модель структуры m_3. В m_3 присутствуют как элементы, соответствующие элементам моделей представления из Ω_1, так и элементы, соответствующие элементам моделей представления из Ω_2. Следовательно, для каждого из элементов любой модели представления $\omega \in \Omega_1 \cup \Omega_2$, в модели структуры m_3 возможно найти отдельный соответствующий элемент. Следовательно, $\exists \eta_3$: $(\Omega_1 \cup \Omega_2) \subset \eta_3(m_3)$, а $\eta_3(m_3) \subset W$. Следовательно, $(\Omega_1 \cup \Omega_2) \subset W$.

Из того, что для $\forall\Omega_1 \subset W$, $\Omega_2 \subset W$ верно 1) $\overline{\Omega_1} \subset W$, $\overline{\Omega_2} \subset W$, 2) $(\Omega_1 \cup \Omega_2) \subset W$, по определению алгебры следует, что W — алгебра пространства Ω.

Так как класс множеств W — алгебра, то по свойству алгебр, можно утверждать следующее:

Следствия теоремы 1

1. Все пространство Ω принадлежит алгебре W.
2. Объединение любого конечного числа множеств $\Omega_1, \ldots, \Omega_n \in W$ принадлежит W.
3. W содержит все конечные пересечения входящих в него множеств.
4. Для $\forall\Omega_1,\Omega_2 \subset \Omega$ $\Omega_1 \setminus \Omega_2 \subset \Omega$, $\Omega_2 \setminus \Omega_1 \subset \Omega$ и $(\Omega_1 \setminus \Omega_2) \cup (\Omega_2 \setminus \Omega_1) \subset \Omega$. Следовательно, W замкнута относительно операций дополнения, разности и конечных объединений и пересечений.

Из следствий теоремы 1 следуют важные практические выводы:

1. Для любой модели структуры возможно построить хотя бы одну модель представления.
2. Для любого набора моделей представления найдется хотя бы одна модель структуры, по которой они могут быть построены.
3. Если существуют модели структуры $m_1, \ldots, m_k$, для которых возможно построить один конечный набор моделей представления, найдется такое преобразование η и модели структуры $m_n, \ldots, m_p$, что результатом этого преобразования будет тот же набор моделей представления.

На основании этих следствий можно утверждать, что при создании и хранении электронных форм следует разделять модели представления и модели содержания, т. к. одни и те же модели представления возможно использовать для различных моделей структур электронных форм.

Пространство моделей представления

Для детального рассмотрения моделей представления определим некоторые отношения, связанные с пространством Ω моделей представления.

Характеристическая функция модели представления

Пусть $\omega \in \Omega$. Рассмотрим элементы ω, которые подчиняются непосредственно корневому узлу графа модели представления. Пусть среди этих узлов n^{ω}_{S} — количество контейнеров, n^{ω}_{A} — количество повторяющихся контейнеров, n^{ω}_{E} — количество терминальных элементов. Определим характеристическую функцию модели представления следующим образом: $\lambda(\omega) = \langle n^{\omega}_{S}, n^{\omega}_{A}, n^{\omega}_{E}\rangle$. Областью значений этой характеристической функции является множество кортежей, образующих пространство Λ. Определим на Λ отношение порядка следующим образом:

$\lambda_1<\lambda_2$ верно, если ($n^1_S < n^2_S$) или ($n^1_S = n^2_S$ и $n^1_A < n^2_A$) или ($n^1_S = n^2_S$ и $n^1_A = n^2_A$ и $n^1_E < n^2_E$),

$\lambda_1 = \lambda_2$ верно, если $n^1_S = n^2_S$ и $n^1_A = n^2_A$ и $n^1_E = n^2_E$.

Рассмотрим кортежи $\langle n^{\omega}_{S}, n^{\omega}_{A}, n^{\omega}_{E}\rangle$ как векторы $(n^{\omega}_{S}, n^{\omega}_{A}, n^{\omega}_{E})$. На Λ введем операции сложения и умножения на число следующим образом: сумма элементов определяется как произведение векторов, а умножение на число — возведение в степень каждого из элементов вектора. Следовательно, Λ является линейным пространством.

Следовательно, на Λ можно ввести скалярное произведение: $(\langle n^1_S, n^1_A, n^1_E\rangle, \langle n^1_S, n^1_A, n^1_E\rangle) = n^1_S\, n^2_S + n^1_A\, n^2_A + n^1_E\, n^2_E$.

Следовательно, Λ можно нормировать: для $\forall\lambda \in \Lambda\ \|\lambda\| = \sqrt{(\lambda,\lambda)}$. Таким образом, Λ — евклидово пространство по определению. Также на Λ можно задать метрику: $d(\lambda_1,\lambda_2) = \|\lambda_1 - \lambda_2\|$.

Операции над моделями представления

Пусть $\omega_1,\omega_2 \in \Omega$. Рассмотрим модель представления $\omega_3 \in \Omega$, граф которой получили из ω_1 путем включения ω_2 (без корня) в какой-либо из контейнеров графа ω_1 (включая корень ω_1). Будем говорить, что $\omega_3 = \omega_1 \cup \omega_2$. Заметим, что при этом не всегда $\omega_1 \cup \omega_2 = \omega_2 \cup \omega_1$. Соответственно, $\omega_3\backslash\omega_2$ — граф, полученный из ω_3, из которого удалили подграф, совпадающий с ω_2. Если в ω_3 нет такого под графа, то $\omega_3\backslash\omega_2 = \varnothing$. Определим дополнение $\overline{\omega}$ как множество всевозможных графов из Ω, отличных от ω. Пересечение моделей представления: $\omega_1 \cap \omega_2$ — граф, который можно выделить как из ω_1, так и из ω_2, т. е. $\omega_4 = \omega_1 \cap \omega_2 \Leftrightarrow \omega_1\backslash\overline{\omega_4} = \omega_2\backslash\overline{\omega_4} = \omega_4$.

Определение 9

Тип объекта называется элементарным, если ни один внутренний узел типа S этого типа объекта не содержит ни одного подчиненного узла. Множество таких типов будем обозначать T_e.

Утверждение 5

Модель структуры можно представить в виде объединения набора элементарных типов и дополнительных ребер, связывающих эти типы, причем набор типов при таких представлениях, будет единственен.

Доказательство

Модель структуры $m \in M$ можно представить следующим образом: $m = t_1 \cup \ldots \cup t_n \cup l_1 \cup l_k$, где $t_i \in D_{min}$ составляют набор типов, $l_i = (\sigma_i, \lambda_j)$ — дополнительные ребра, соединяющие σ_i и λ_j, где σ_i —терминальный узел типа S одного из типов набора, а λ_j — корень t_j. Если t_j не является элементарным, то его, т. к. он также является моделью структуры, можно представить в виде объединения других типов и дополнительных ребер: $t_j = t_h \cup \ldots \cup t_g \cup l_s \cup l_z$. Подобным образом типы можно дробить до элементарных типов. То есть существует набор элементарных типов $t_r \ldots t_k$ такой, что $m = t_r \cup \ldots \cup t_p \cup l_m \ldots \cup l_h$.

Пусть теперь $m = t_r \cup \ldots \cup t_p \cup l_m \cup l_h = t_q \cup \ldots \cup t_u \cup l_s \ldots \cup l_g$. Следовательно, в указанных наборах типов существует t_i: $t_i = t_a \cup \ldots \cup t_b \cup l_c \ldots \cup l_d$. Это невозможно, т. к. t_i является элементарным типом. Получили противоречие.

Определение 10

Все модели представления ω: $\omega \in \eta(t)$, $t \in T_e$, будем называть элементарными. Множество элементарных моделей представления обозначим Ω_e.

Утверждение 6

Элементарная модель представления является простым семантическим блоком.

Теорема 2

Модель представления любого объекта можно представить как объединение элементарных моделей представления.

Доказательство

Как было доказано выше, любую модель структуры можно представить в виде набора элементарных типов и ребер, связывающих эти типы. То есть $m = t_1 \cup \ldots \cup t_n \cup l_1 \cup l_k$, где t_i — элементарные типы. Следовательно, т. к. для $\forall t_i \in D\ \exists \eta_i$, $\omega_i \in \Omega$: $\omega_i = \eta_i(t_i)$, и все ω_i принадлежат алгебре W, то представления $\Omega_m = \eta_p(t_p) \cup \ldots \cup \eta_r(t_r) \subset \Omega$. $\eta_i(t_i)$ — множество элементарных моделей представления по определению.

Определение 11

Элементарные модели представления назовем эквивалентными, если их характеристические функции равны. $\omega_1, \omega_2 \in \Omega_e$, $\lambda(\omega_1) = \lambda(\omega_2) \Rightarrow \omega_1 \sim \omega_2$.

Это определение законно, т. к. для характеристических функций моделей представления выполняются все свойства отношения эквивалентности.

Пусть $\omega = \omega_1 \cup \ldots \cup \omega_n$, где $\omega_1, \ldots, \omega_n$ — элементарные модели представления. Тогда $\lambda(\omega) = \lambda(\omega_1 \cup \ldots \cup \omega_n)$. Введем обобщенную характеристическую функцию модели представления следующим образом: $\overline{\lambda}(\omega) = \lambda(\omega_1) + \ldots + \lambda(\omega_n)$.

Определение 12

Обобщенной характеристической функцией модели представления называется множество значений характеристических функций элементарных моделей представления, объединением которых является модель представления. То есть для $\forall\omega \in \Omega$, $\forall\omega_1, \ldots,\omega_n \in \Omega_e$: $\omega = \omega_1 \cup \ldots \cup \omega_n$, $\overline{\lambda}(\omega) = \{\lambda(\omega_1), \ldots,\lambda(\omega_n)\}$.

Пусть t — элементарный тип объекта. Пусть $\{\omega_{ti}\} = \eta(t)$, $i = 1,\ldots,n$ — множество элементарных моделей представления объектов типа t. Тогда $\overline{\lambda}(\omega_{ti\ max}) = \overline{\lambda}(\omega_{tj\ max})$ для $\forall i, j$.

Определение 13

Модели представления назовем эквивалентными, если их обобщенные характеристические функции равны. $\omega_1,\omega_2 \in \Omega$, $\overline{\lambda}(\omega_1) = \overline{\lambda}(\omega_2) \Rightarrow \omega_1 \sim \omega_2$.

Аналогично можно ввести определения характеристической функции μ для модели структуры объекта.

Утверждение 7

Пусть $t \in T_e$, $\omega \in \Omega_e$, $\omega \in \eta(t)$. Тогда $\lambda(\omega) = \mu(t)$. То есть значение характеристической функции элементарного типа равно значению характеристической функции модели представления, если они принадлежат одной модели формы.

Так как характеристические функции элементарных моделей представления и типов равны, то обобщенная характеристическая функция модели представления будет состоять из значений характеристических функций элементарных типов, составляющих модель структуры. Следовательно, верно следующее утверждение:

Утверждение 8

Пусть $m = t_1 \cup \ldots \cup t_n \cup l_1 \cup l_k$, $t_i \in D_{min}$, $\omega \in \Omega$, $\omega_1, \ldots,\omega_n \in \Omega_e$, $\omega = \omega_1 \cup \ldots \cup \omega_n$, $\omega \in \eta(m) \Rightarrow \overline{\lambda}(\omega) = \{\mu(t_p), \ldots, \mu(t_r)\}$, где $p \in \{1\ldots n\}, \ldots, r \in \{1\ldots n\}$.

Следовательно, набор различных эквивалентных моделей представления может быть использован для представления соответствующих объектов, удовлетворяющих одной модели структуры. Следовательно, набора всех моделей представления, эквивалентных некоторой $\omega \in \Omega$, достаточно для представления всех объектов, удовлетворяющих моделям структур, состоящим из элементарных типов, характеристические функции которых принимают значения из множества $\overline{\lambda}(\omega)$.

Таким образом, если есть модель структуры, то для построения моделей представления, представляющих всевозможные объекты, удовлетворяющие этой модели структуры, нужно представить модель структуры в виде объединения элементарных типов, для каждого из этих типов построить элементарную модель представления, а затем построить всевозможные объединения этих моделей представления.

Очевидно, что строить всевозможные объединения моделей представления на практике отнимет достаточно много времени. Для того чтобы этого не делать, в множество W добавим элемент, означающий условный контейнер. Этот элемент позволит строить объединение двух моделей представления в зависимости от того, какой объект отображается с помощью модели, содержащей этот элемент. Следовательно, если вместо каждого контейнера в моделях представления вставлять условный контейнер, то нужное объединение моделей представления может быть построено автоматически, например, перед тем, как представить какой-либо объект, который удовлетворяет модели структуры, имеющей связи с этими моделями представления. Условный контейнер назовем вложенной динамической миниформой.

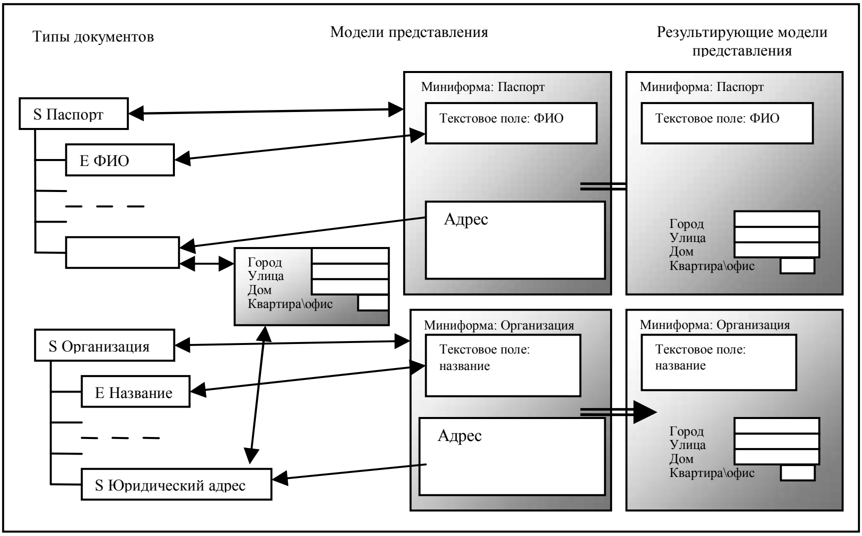


Рис. 1

Использование условных контейнеров также позволяет использовать одни и те же модели представления для представления эквивалентных частей различных моделей представления. Например, на рис. 1 показано, как можно построить модель представления для представления структуры

«Адрес» и использовать эту модель представления в составе модели представления для документов «Паспорт», «Организация», «Письмо» и других, для которых требуется вводить или отображать какой-либо адрес. При отображении каждого из указанных документов будет использована отдельная модель представления, но вместо пустого блока для адреса подставится одна и та же модель представления для адреса.

Также условный контейнер может позволить отображать внутри себя различные модели представления в процессе работы с данными формы. Например, в зависимости от введенных данных, внутри контейнера можно увидеть поля модели представления для параметров физического лица или для параметров юридического лица.

Отображение электронной формы на устройствах вывода

До сих пор рассматривались проблемы состава элементов электронной формы и связи внутри формы. Но модель представления электронной формы предназначена для отображения объекта на устройствах вывода, поэтому необходимо определить правила создания соответствующих отображений модели представления. В общем случае модель представления не определяет способ ее отображения на конкретных устройствах или для конкретных нужд. Поэтому различные реализации части процессора форм, отвечающей за отображение формы, могут создавать на устройствах вывода различные отображения одних и тех же моделей представления. Например, форма обычно имеет различный внешний вид при выводе формы на экран и при выводе на принтер. При этом основные параметры элементов формы должны учитываться везде. Такими параметрами могут быть параметры расположения элементов, шрифт, цвет и некоторые другие. Кроме того, при выводе формы необходимо учитывать для каких целей она отображается: для редактирования данных, для подготовки формы к печати, для просмотра без возможности редактирования данных или для просмотра и частичного редактирования данных объекта.

Для того чтобы иметь возможность указать необходимые параметры отображения формы, расширим пространство W, добавив еще два измерения:

1. Параметры расположения элементов на форме.
2. Параметров, определяющие на внешний вид элементов и инструменты редактирования данных, которые они отображают.

Расширенное таким образом пространство W обозначим как W^V. При этом существует проекция P_W: $W^V \rightarrow W$. Проекция получается из W^V отбрасыванием введенных измерений.

Расположение элементов формы на устройствах вывода

В этой главе отображение элемента модели представления на устройстве вывода будем называть **элементом формы**. Причем в понятие «элемент формы» будем включать отображение любого элемента пространства W на устройстве вывода. Отображение контейнера на устройстве вывода будем называть контейнером и обозначать $C(B_i)$, где B_i — семантический бок, отображаемый контейнером.

Каждый элемент формы устроен следующим образом (рис. 2):

1. Задана некоторая область, которая отображается на устройстве вывода заданным способом. Эту область будем называть **областью отображения** элемента формы.
2. Задана область, на которой отображаются данные элемента. Эту область будем называть **областью отображения данных** элемента формы, или просто областью данных.
3. Задано расположение данных на области отображения данных.
4. Внутри области отображения отображается часть области отображения данных.
5. Область отображения данных подвижна относительно области отображения.

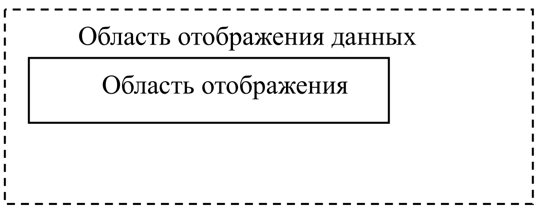


Рис. 2

Следовательно, если область отображения данных меньше области отображения элемента формы, то на устройстве вывода всегда отображается только часть области отображения данных, поэтому элементы формы предоставляют способы смещения области отображения данных относительно области отображения для того, чтобы имелась возможность отобразить требуемую часть данных элемента формы на устройстве вывода.

Расположение элементов электронной формы возможно указывать несколькими способами:

- указанием координат каждого элемента формы;
- указанием сдвига элементов относительно друг друга;
- указанием начальных координат и параметров сдвига элементов.

При этом могут быть указаны параметры, определяющие конфигурацию элемента на поле формы.

Для указания координат элемента используется набор чисел $P = \langle x, y, w, h, \ldots \rangle$, пара из которых означает абсолютные координаты некоторой точки p на некоторой координатной плоскости. К точке p жестко привязывается какая-либо точка элемента управления (левый верхний угол прямоугольника или центр представления элемента произвольной конфигурации). Остальные числа из P могут указывать на размеры представления элемента, сдвиги точек представления относительно p, абсолютные координаты других точек представления элемента и т. п. Обычно используются формы с прямоугольными элементами, координаты которых задают левый верхний и правый нижний углы прямоугольника, т. е. $P_п = \langle x_1, y_1, x_2, y_2 \rangle$. Реже прямоугольники используются для указания пространства, которое должен максимально заполнить некоторое произвольное изображение элемента, созданное в графическом редакторе.

Указание сдвигов элементов может использоваться в том случае, если нет необходимости оговаривать размеры элементов и при изменении размеров всей формы, или при вводе данных в форму, необходимо пропорционально изменять размеры элементов внутри формы так, чтобы расстояние между ними оставалось постоянным или изменялось по какому-либо закону, например, умножалось бы на некоторый коэффициент. При указании подобных параметров может возникнуть ситуация, когда некоторые параметры влияют на все элементы некоторого контейнера формы. Такие параметры следует относить к параметрам расположения контейнера, содержащего эти элементы. Пусть B_i — семантический блок [3], тогда расположение этого блока на плоскости можно указать в виде кортежа: $P(B_i) = \langle \Delta P(B_p), \Delta\ P(B_r), \Delta\ P(B_t), P(\Delta B_q) \rangle$. Элементами этого кортежа являются значения сдвигов относительно элементов B_p, B_r, B_t, B_q соответственно по горизонтали слева и справа, и по вертикали сверху и снизу.

При указании начальных координат и параметров сдвига элементов производится смешение двух предыдущих способов указания параметров расположения элементов. В этом случае могут быть жестко заданы координаты некоторых элементов, а для остальных — параметры сдвига относительно элементов.

Итак, в общем случае расположение элементов на форме задается набором чисел $P = \langle x, y, w, h, \ldots \rangle$, трактовка которых зависит от реализации процессора форм или от параметров формы, указывающих способы трактовки P.

Элементы модели представления могут объединяться в семантические блоки [3], которые могут иметь свои отдельные параметры расположения. В модели представления семантические блоки можно выделить в виде элементов множества W_φ, каждый из которых имеет свои параметры отображения.

Итак, позиционирование элементов можно свести к позиционированию семантических блоков и элементов (которые также могут являться семантическими блоками) внутри семантических блоков.

При проектировании модели представления формы часто возникает ситуация, когда невозможно или нежелательно отображение всех данных формы одновременно. Подобная ситуация возникает, если:

1. Данные невозможно уместить на доступном пространстве.
2. Одновременное отображение всех данных не удобно для восприятия пользователем [4].
3. Часть данных не актуальна.

Поэтому возникает необходимость скрыть часть данных совсем или открывать их по желанию пользователя. При открытии данных по желанию пользователя допустима ситуация, когда часть остальных данных скрывается или частично перекрывается запрошенными данными. В силу перечисленных ограничений, запрашиваемые пользователем данные, также могут быть частично скрыты. При этом пользователь должен иметь возможность просмотреть или отредактировать все доступные ему данные формы. Следовательно, для доступа к скрытым данным, необходимо иметь способ доступа к ним с помощью элементов навигации, связанных с элементами просмотра и\или редактирования данных. Элементы навигации можно разделить на два типа:

1) дискретного доступа к данным;
2) непрерывного доступа к данным.

Например, элемент прокрутки является элементом навигации непрерывного доступа, а набор закладок — элементом навигации дискретного доступа. При дискретном доступе имеет смысл разделить часть данных формы на однотипные семантические блоки и осуществлять доступ к каждому из этих блоков отдельно. Часто элементы навигации используются для смещения области отображения данных элементов формы относительно области отображения элементов формы.

При сокрытии части данных формы, скрываются и элементы, отображающие эти данные. При этом на поле формы могут возникнуть обширные пустые пространства. Для того чтобы избежать подобной ситуации, необходимо предусмотреть возможность заполнения пустого пространства видимыми элементами формы. Для этого необходимо задать законы перемещения элементов формы внутри своего контейнера. Подобные законы могут быть заданы в процессоре форм или формально описаны параметрами элементов модели представления. При указании расположения элементов формы в виде сдвигов элементов относительно друг друга, проблема заполнения пустых пространств будет решена автоматически.

Кроме скрывания данных формы, часто возникает необходимость уменьшать или увеличивать размеры элементов формы в зависимости от наличия и объема данных в элементах формы. Например, если на форме две таблицы расположены одна под другой, и в верхней таблице данных нет, то естественно большую часть пространства, занимаемого этими таблицами, заполнить нижней таблицей. Подобным образом налаживается взаимодействие любых элементов и блоков формы, которые разрешено сужать или расширять. Для того чтобы определить законы, по которым изменяются размеры элементов формы, также необходимо задать некоторое количество соответствующих параметров. Обычно такие параметры обозначают направление, в котором элементы формы могут изменять размеры и предельные значения, за рамки которых нельзя выводить размеры элементов формы.

При отображении на форме массива данных, необходимо учитывать, что массив является списком однотипных структур. Поэтому массив отображается блоками, имеющими одинаковую модель представления, но не всегда имеющими одинаковый размер (в одном элементе все поля актуальны, а в другом какие-то поля не актуальны и скрыты, вследствие чего уменьшен размер второго элемента). При отображении массива используется отдельный контейнер, содержащий однотипные блоки, которые также могут являться контейнерами. Количество элементов массива заранее не указывается и их невозможно разместить заранее вручную при проектировании формы. Поэтому внутри контейнера, отображающего массив, необходимо определить закон позиционирования элементов массива. Так как в массивах часто важен порядок следования элементов, то для описания закона позиционирования элементов, в контейнере необходимо указать позицию первого элемента массива, направление, в котором будут последовательно выводиться элементы массива, интервалы между позициями соседних элементов и, не обязательно, предельные значения позиций элементов. Обычно элементы массива располагаются следующими способами:

1. Таблица. Подробное описание представления табличных данных можно найти в [4].
2. Вертикальный список представлений элементов массива.
3. Горизонтальный список представлений элементов массива.

Вертикальный список является аналогом столбцовой таблицы [4], в ячейках которой отображаются элементы массива. При этом, если достигается предельное значение позиций элементов массива по вертикали, следующий элемент отображается на одной горизонтали с первым элементом массива с указанным отступом от позиции первого элемента вправо (обычно) или влево. Горизонтальный список является аналогом строчной таблицы [4], в ячейках которой отображаются элементы массива. При этом, если достигается предельное значение позиций элементов массива по горизонтали, следующий элемент отображается на одной вертикали с первым элементом массива с указанным отступом от позиции первого элемента вниз (обычно) или вверх.

Пусть параметры расположения элементов формы задаются кортежем чисел $P_n = \langle p_1, p_2, \ldots, p_n \rangle$, где n — константа, определяемая реализацией процессора форм. Множество всевозможных кортежей вида P можно объединить в пространство параметров отображения элементов формы. Обозначим его как W_P. Графическим представлением этого пространства являются изображения границ элементов форм. С учетом вышесказанного, границы элементов могут как угодно пересекаться или совпадать.

Пусть имеется k-мерное устройство вывода. Пространство вывода для этого устройства обозначим V_k. Пространство V_k состоит из множества точек. Форма на этом устройстве вывода представляется в виде замкнутой области, включающей множество других замкнутых областей, которые являются элементами формы. Положение и размеры этих областей заданы элементами пространства W_p. Пусть также множеством параметров $C = \{c_1, c_2, \ldots, c_q\}$ задана конфигурация областей. Всевозможные параметры, задающие конфигурацию, выделим в пространство W_c. Следовательно, для корректного отображения элементов модели представления на область формы, необходимо существование взаимно однозначного соответствия $(W_p \cup W_c) \to V_k$.

Обычно устройства вывода являются двумерными (например, принтер или экран монитора). Поэтому далее будем рассматривать двумерное пространство V_k, обозначаемое как V_2. Расположение элементов формы можно задавать прямоугольниками, т. к. наиболее наглядной формой элементов формы является прямоугольник, а расположение не прямоугольного элемента формы возможно определить прямоугольником, в который вписан элемент формы. Кроме того, не пересекающимися прямоугольниками можно полностью покрыть пространство электронной формы, которое обычно является прямоугольным, т. е. можно более плотно заполнить имеющуюся область. Таким образом, далее везде расположение элемент формы будет задаваться прямоугольниками.

Для позиционирования элементов формы выберем декартову систему координат, связанную с левым верхним углом формы. Причем ось ординат направлена вертикально вниз, а ось абсцисс — горизонтально вправо. Далее везде будем использовать такую систему координат для указания координат элементов формы. Если элемент формы обозначается как e, то координаты прямоугольника его отображения будем обозначать $(e_{x1}, e_{y1}, e_{x2}, e_{y2})$.

При работе с формой часто возникают следующие задачи, требующие решения:

1. Размещение элементов массива внутри контейнера, отображающего этот массив. Эта задача требует решения при выводе массивов на экран.
2. Определение того, является ли скрытым или нет определенный элемент формы и позиционирование элементов формы так, чтобы он стал видим. Эта задача требует решения в том случае, если пользователю необходимо показать конкретный элемент данных, элемент формы для которого может быть скрыт другими элементами формы.

Рассмотрим задачу 1 размещения элементов массива в контейнере. Для указания координат возьмем описанную ранее систему координат, связанную с левым верхним углом контейнера, отображающего массив (рис. 3).

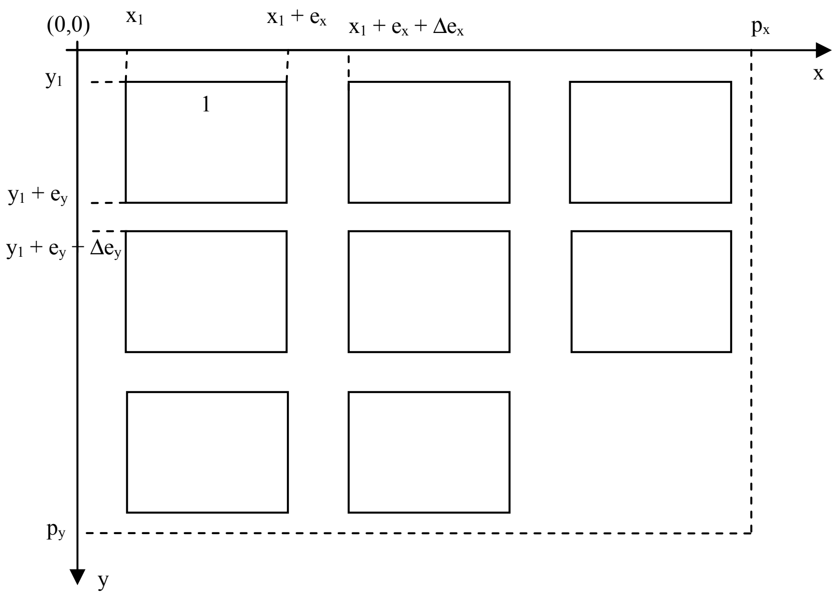


Рис. 3

Пусть даны следующие параметры:

- Sc = (w, h) — размеры контейнера, отображающего массив, заданы парой чисел: ширина и высота;
- Se = (e_x, e_y) — размер элемента формы, отображающего элемент массива в данном контейнере;
- Δe = (Δe_x, Δe_y) — отступ между элементами массива по горизонтали и по вертикали соответственно;

- $Pv = (p_x, p_y)$ — предельные значения: максимальные размеры области отображения массива по горизонтали и по вертикали;
- $Pf = (x_1, y_1)$ — позиция первого элемента массива задана координатами относительно левого верхнего угла контейнера;
- $b = (b_x, b_y)$ — координаты единичного вектора, задающего направление, в котором будут последовательно выводиться элементы массива.

Тогда координата левого верхнего угла i-го элемента массива будет определяться по формуле:

$$\begin{gathered}(x_i, y_i) = Pose(Se, \Delta e, Pv, Pf, b, i) = \\ = (x_1 + [(i - ((i-1) \operatorname{div} k_x))k_x - 1)b_x + (i \operatorname{div} k_y)b_y](e_x + \Delta e_x), \\ y_1 + [(i - ((i-1) \operatorname{div} k_y))k_y - 1)b_y + (i \operatorname{div} k_x)b_x]\,(e_y + \Delta e_y),\end{gathered} \tag{1}$$

где

$$k_x = (p_x - x_1) \operatorname{div} (e_x + \Delta e_x), \qquad k_y = (p_y - y_1) \operatorname{div} (e_y + \Delta e_y).$$

Заметим, что если какое-либо из предельных значений p_α не задано, то будем считать что $p_\alpha \to \infty$. При этом $k_\alpha = 0$.

Если $b = (1, 0)$, то элементы располагаются горизонтально построчно (рис. 4, а). Если $b = (0, 1)$, то элементы располагаются вертикально по столбцам (рис. 4, б). Если $\Delta e = (0, 0)$, $Pv = (\infty, \infty)$, $Pf = (0,0)$, $b = (0, 1)$, то получим отображение массива в виде обыкновенной столбцовой таблицы или строчной таблицы при $b = (1, 0)$. Так как в таблице $\Delta e = (0, 0)$, то отображение массива в виде таблицы является наиболее компактным и наиболее часто используемым.

Определение 14

Скрытым будем считать такой элемент формы, что не все точки, принадлежащие прямоугольнику, описывающему положение этого элемента, отображены устройством вывода.

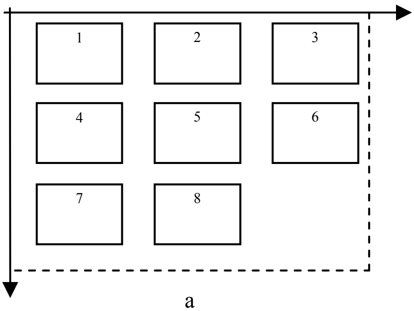
Определение 15

Элемент формы, не являющийся скрытым, будем называть видимым элементом формы.

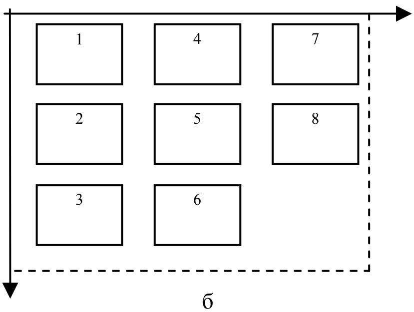
Рассмотрим задачу 2 определения того, является ли скрытым или нет определенный элемент e формы. Если элемент скрыт от пользователя сознательно в силу его неактуальности, то решение очевидно. Попробуем найти решение, если элемент может быть скрыт из-за того, что он перекрывается другими элементами формы.

Рассмотрим положение элемента e формы в виде множества точек, принадлежащих области, ограниченной прямоугольником области отображения элемента формы с координатами (e_{x1}, e_{y1}, e_{x2}, e_{y2}). Это множество обозначим как Q(e). Таким образом,

$$Q(e) = \{q = (q_x, q_y) \in V_2: q_x \geq e_{x1}, q_x \leq e_{x2}, q_y \geq e_{y1}, q_y \leq e_{y2}\}. \qquad (2)$$



а



б

Рис. 4

Если элемент формы e1 перекрывает элемент формы e2, то $Q(e1) \cap Q(e2) \neq \varnothing$. Пересечение элементов формы e1, …, en обозначим как Q(e1, …, en). Тогда, при n = 2,

$$Q(e1, e2) = Q(e1) \cap Q(e2) =$$
$$=\{q = (q_x, q_y) \in V_2: q_x \geq e1_{x1}, q_x \leq e1_{x2}, q_x \geq e2_{x1}, q_x \leq e2_{x2},$$
$$q_y \geq e1_{y1}, q_y \leq e1_{y2}, q_y \geq e2_{y1}, q_y \leq e2_{y2}\}.$$

Соответственно, не перекрываемая область элемента формы состоит из точек $(Q(e1) \cup Q(e2))/(Q(e1) \cap Q(e2))$.

Определение 16

Площадью элемента формы e назовем площадь на отображении формы, покрываемую множеством Q(e). Площадь будем обозначать SQ(e).

Определение 17

Пусть e1, …, en — элементы формы. Площадь SQ(e1, …, en) назовем максимальной площадью пересечения элементов e1, …, en, если не существует таких координат e1', …, en' элементов формы, что

SQ(e1, …, en) < SQ(e1', …, en'). Максимальную площадь будем обозначать $S_{max}Q$(e1, …, en).

Рассмотрим модель представления, имеющую ранг r > 1 [3]. Пусть тогда контейнер, соответствующий семантическому блоку B_i^r, содержит элемент формы, который требуется сделать видимым.

Для постепенного рассмотрения перекрывания элементов формы сначала на расположение элементов формы наложим ограничение: элементы формы не могут выходить за пределы своего контейнера и элементы, принадлежащие одному контейнеру, не могут перекрывать друг друга. В этом случае задача сводится к поиску пересечений элемента со всеми контейнерами, внутри которых находится элемент e. И в случае, если такие пересечения найдены, произвести смещение областей отображения данных элементов формы относительно областей отображения этих элементов таким образом, чтобы открыть максимальную область отображения элемента e внутри области отображения контейнеров, содержащих эти элементы.

Утверждение 9

Пусть для любых элементов формы e1, e2, e1 $\in$ $C(B_i^k)$, e2 $\in$ $C(B_i^k)$,

$Q(e1) \cap Q(e2) = \varnothing$ для любых i, k и $Q(e1) \cap Q(C(B_i^k)) \neq \varnothing$, $Q(e2) \cap Q(C(B_i^k)) \neq \varnothing$. Пусть e $\in$ $C(B_J^m)$ и $Q(e) \not\subset Q(C(B_J^m))$. Тогда для $C(B_J^m)$ $\exists$ смещение координат области отображения данных $\Delta e = (\Delta e_x, \Delta e_y)$ относительно начала координат формы, такое, что $SQ(e + \Delta e, C(B_J^m)) = S_{max}Q(e, C(B_J^m)) > 0$. При этом $\Delta e = \{(\Delta e_x, \Delta e_y)$:

$$C(B_J^m)_{x1} - e_{x1} \leq \Delta e_x \leq C(B_J^m)_{x2} - e_{x2},$$
$$C(B_J^m)_{y1} - e_{y1} \leq \Delta e_y \leq C(B_J^m)_{y2} - e_{y2}\}.$$

Доказательство

Очевидно, одним из значений Δe будет $\Delta e^o = (C(B_j^m)_{x1} - e_{x1}, C(B_j^m)_{y1} - e_{y1})$, при котором $SQ(e + \Delta e^o, C(B_j^m)) = S_{max}Q(e, C(B_j^m))$. При этом Δe^o будет единственным значением Δe, если $Q(e + \Delta e^0, C(B_j^m)) = Q(C(B_j^m))$. См. рис. 5.

Рассмотрим сдвиг по оси Ox: при $e_{x1} + \Delta e^o_x = C(B_j^m)_{x1}$ — совпадают левые верхние углы прямоугольников элемента и контейнера. Область данных контейнера будем сдвигать до того момента, пока не совпадут правые верхние углы прямоугольников: $e_{x2} + \Delta e^1_x = C(B_j^m)_{x2}$. Получим, что

$\Delta e^1{}_x = C(B_j{}^m)_{x2} - e_{x2}$. Аналогично для оси Oy — от совпадения левых верхних углов до совпадения левых нижних. При сдвиге области данных в пределах $\Delta e^o{}_x \le \Delta e^2{}_x \le \Delta e^1{}_x$ и $\Delta e^o{}_y \le \Delta e^2{}_y \le \Delta e^1{}_y$ площадь $SQ(e + \Delta e^o, C(B_j{}^m))$ будет максимальной и больше нуля. □

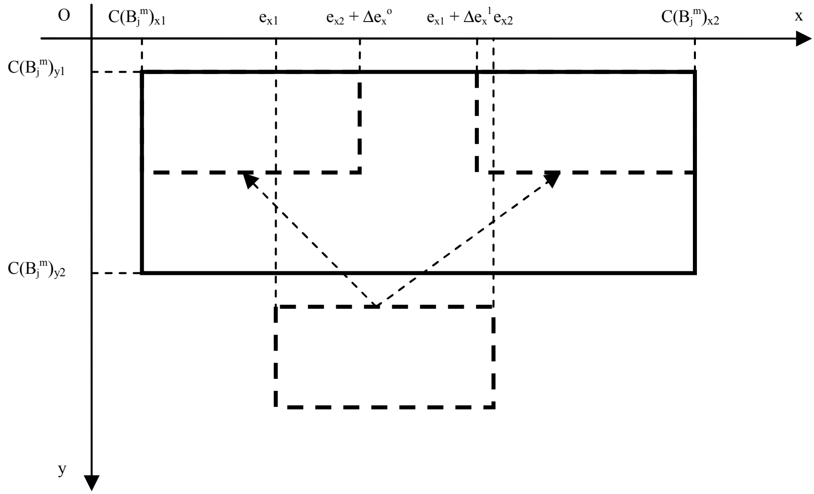


Рис. 5

Если $e \in C(B_j{}^m)$ и $C(B_j{}^m) \subset C(B_j{}^{m-1})$, $Q(e) \subset Q(C(B_j{}^m))$ и $Q(e) \not\subset Q(C(B_j{}^{m-1}))$, то элемент e будет скрытым или частично скрытым. Согласно утверждению 1 существует сдвиг $\Delta C(B_j{}^m)$ области данных контейнера $C(B_j{}^{m-1})$, такой что,

$SQ(C(B_j{}^m) + \Delta C(B_j{}^m), C(B_j{}^{m-1}) = S_{max}Q(C(B_j{}^m) + \Delta C(B_j{}^m), C(B_j{}^{m-1})$. При реализации указанных сдвигов получим: $SQ(e + \Delta e, C(B_j{}^m) + \Delta C(B_j{}^m), C(B_j{}^{m-1})) = S_{max}Q(e, C(B_j{}^m), C(B_j{}^{m-1}))$. Аналогично можно показать, что существуют сдвиги координат областей данных контейнеров $C(B_1{}^1)$, …, $C(B_1{}^m)$, такие, что $SQ(e + \Delta e, C(B_1{}^1), C(B_i{}^2) + \Delta C(B_i{}^2), \ldots, C(B_j{}^m) + \Delta C(B_j{}^m)) = S_{max}Q(e, C(B_1{}^1), \ldots, C(B_j{}^m))$. Следовательно, элемент e может быть отображен на устройстве вывода внутри каждого из прямоугольников, принадлежащих $Q(C(B_1{}^1))$, площадь которых не больше, чем $S_{max}Q(e, C(B_1{}^1), \ldots, C(B_j{}^m))$.

Если $SQ(e) > S_{max}Q(e, C(B_1{}^1), \ldots, C(B_j{}^m))$, то на устройстве вывода можно будет получить только часть Q(e), т. е. e будет частично скрытым всегда. В этом случае необходимо отображать на экране ту часть Q(e), которая актуальна в данный момент.

Определение 18

Пусть точка $q \in Q(e)$. Q назовем актуальной точкой элемента e, если в текущий момент времени на устройстве вывода необходимо отображать данные элемента e, которые отображаются внутри области $\{a = (a_x, a_y): a_x \geq q_x, a_y \geq q_y\}$.

Очевидно, что при наличии актуальной точки q элемента e, значениями Δe, в условиях утверждения 1, будут:

$$C(B_j^m)_{x1} - q_{x1} \leq \Delta e_x \leq C(B_j^m)_{x2} - q_{x2},\ C(B_j^m)_{y1} - q_{y1} \leq \Delta e_y \leq C(B_j^m)_{y2} - q_{y2}.$$

Если для элемента формы e не задана актуальная точка q, будем считать, что $q = (e_{x1}, e_{y1})$. Таким образом, при наличии актуальной точки, всегда найдутся такие сдвиги координат областей данных контейнеров $C(B_1^1)$, …, $C(B_1^m)$, что на устройстве вывода отобразится актуальная часть данных элемента e, ограниченная площадью $S_{max}Q(e, C(B_1^1), \ldots, C(B_j^m))$.

Теперь рассмотрим ситуацию, при которой элементы формы могут выходить за пределы своего контейнера и элементы, принадлежащие одному контейнеру, могут перекрывать друг друга. В этом случае должна существовать возможность автоматически перемещать элементы формы, которые могут перекрывать собой другие элементы. В противном случае нарушается требование о доступности всех данных формы и форма считается некорректной. Следовательно, найдя сдвиги координат областей данных контейнеров $C(B_1^1)$, …, $C(B_1^m)$, образующие $Q(e, C(B_1^1), \ldots, C(B_j^m))$: $SQ(e + \Delta e, C(B_1^1), C(B_i^2) + \Delta C(B_i^2), \ldots, C(B_j^m) + \Delta C(B_j^m)) = S_{max}Q(e, C(B_1^1), \ldots, C(B_j^m))$, необходимо определить, какие элементы формы пересекаются с $Q(e + \Delta e, C(B_1^1), C(B_i^2) + \Delta C(B_i^2), \ldots, C(B_j^m) + \Delta C(B_j^m))$ и найти сдвиги этих элементов, чтобы элементы с новыми координатами не пересекали $Q(e + \Delta e, C(B_1^1), C(B_i^2) + \Delta C(B_i^2), \ldots, C(B_j^m) + \Delta C(B_j^m))$.

При отображении данных также следует учесть удобство их чтения на устройстве вывода. Обычно для просмотра или редактирования данного требуется просматривать некоторые соседние данные. Например, при вводе имени человека, удобно видеть его фамилию. Поэтому будем считать, что актуальную точку q элемента $e \in C(B_j^m)$ следует помещать по центру области $Q(C(B_j^m))$. С учетом такой поправки, значение Δe следует выбирать следующим:

$$\Delta e_x = C(B_j^m)_{x1} - q_{x1} + (C(B_j^m)_{x2} - C(B_j^m)_{x1})/2,$$
$$\Delta e_y = C(B_j^m)_{y1} - q_{y1} + (C(B_j^m)_{y2} - C(B_j^m)_{y1})/2.$$

Если при этом $e_{x2} + \Delta e_x < C(B_j^m)_{x2}$ или $e_{y2} + \Delta e_y < C(B_j^m)_{y2}$, то Δe нужно изменить так, чтобы в $Q(C(B_j^m))$ поместилась максимально большая часть $Q(e)$.

Из приведенных рассуждений следует алгоритм отображения актуальных данных любого элемента формы:

- Шаг 1. Определяется актуальная точка q элемента e.
- Шаг 2. Вычисляется сдвиг $\Delta C(B_j^m)$ области данных контейнера $C(B_j^m)$: $e \in C(B_j^m)$.
- Шаг 3. Сдвигается область данных контейнера $C(B_j^m)$ на $\Delta C(B_j^m)$.
- Шаг 4. $e := C(B_j^m)$, $m := m - 1$. Если $m = 0$, то завершаем процесс, иначе — переходим к шагу 1.

Литература

1. *Романов Б. Л.* Представление структурированных информационных объектов в виде электронных форм // Сборник трудов Института системного анализа РАН, 2002.
2. *Пугачев В. С.* Лекции по функциональному анализу. М.: Изд-во МАИ, 1996.
3. *Богачева А. Н.*, *Емельянов Н. Е.* Семантическая модель документа // Системные исследования. Ежегодник 2001. М.: УРСС, 2003. С. 360–375.
4. *Иванов Ю. Н.*, *Емельянов Н. Е.*, *Сотникова Р. А.* Документы: типы, описание. Препринт. Всесоюзный научно-исследовательский институт системных исследований. М., 1987.
5. *Bogacheva A. N.*, *Emeljanov N. E.* Duality between Document Structure and Data Base Structures // ABIS’94 Proceeding of The International Workshop on Advances in Databases and Information Systems, Moscow, May 23–26, 1994.