

Е.С. Москвичев, П.А. Тимофеев, С.Н. Сатышев, А.А. Солнцев

## МЕТОДЫ И МОДЕЛИ ФОРМИРОВАНИЯ АДАПТИВНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ В СИСТЕМАХ ПОДДЕРЖКИ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ

**Аннотация.** В статье рассматриваются вопросы формирования инструментальных средств и методов конструирования адаптивных пользовательских интерфейсов. Предлагается аппарат конструирования интерфейса на основе автоматной схемы разбора. Сформированы требования к графическому редактору конструирования структуры интерфейса и схемы разбора интерактива пользователя. Алгоритмическая структура приложений формализована в виде сетей Петри. Рассмотрены вопросы оптимизации временных характеристик реализации пользовательских запросов за счет динамической кластеризации ресурсов. Предложена формализация в виде Марковской цепи, которая учитывает начальное распределение вероятностей обращения к ресурсам системы.

**Ключевые слова:** интерфейс, адаптация, сценарий, вложенные процессы.

### Введение

В настоящее время на предприятиях промышленности и транспортного комплекса установлены и активно используются нормативно-правовые и информационно-справочные базы данных, реестры лицензиатов и лицензий, базы данных транспортных средств и субъектов транспортной деятельности и другие программные комплексы. Вместе с тем, этот этап можно охарактеризовать только как этап первоначального создания и внедрения разрозненных информационных и телекоммуникационных технологий. Компьютерная подготовка персонала не всегда удовлетворяет постоянно возрастающим требованиям к эффективной эксплуатации автоматизированных систем управления. Более того, один и тот же программный продукт часто предполагает использование различными группами пользователей для реализации конкретных целей. В связи с этим необходимы инструментальные средства создания адаптивных пользовательских интерфейсов, что, несомненно, повысит оперативность принятия решений с учетом уровня подготовленности каждого пользователя и направленности его деятельности. При этом возникает необходимость реализации настройки программных компонентов на различные конфигурации с учетом интерфейсных решений для конкретных пользователей.

Кроме того, необходим учет специфики интерфейсного взаимодействия пользователя, которая может быть обеспечена динамическим перераспределением функционала пользовательского интерфейса. Этой про-

блематике в настоящее время уделяется недостаточно внимания.

### Требования к адаптивным интерфейсам

На основе анализа методов навигации показано, что технические приемы адаптивной поддержки навигации используются для достижения нескольких целей адаптации: обеспечение глобального руководства, обеспечение локального руководства, поддержка локальной ориентации, поддержка глобальной ориентации, помощь в индивидуализации информации. Индивидуализация информации – это новая цель для систем адаптивной гипермедиа [1]. Индивидуализированные поля видимости – это способ организовать электронное рабочее место для пользователей, которым для ежедневной работы необходим доступ к довольно небольшой части всего гиперпространства. Традиционным способом защиты этих пользователей от сложности всего гиперпространства является организация индивидуализированных, ориентированных на цель полей видимости. Каждое поле видимости представляет собой список ссылок ко всем гипермедиа-документам. Традиционно создание своих индивидуализированных полей видимости и управление ими является обязанностью пользователей. Классические гипермедиа-системы и современные WWW-браузеры в качестве способа создания индивидуализированных полей видимости предлагают «закладки» и «горячие списки». Более развитые системы предлагают механизмы адаптивности более высокого уровня, основанные на образах и моделях пользователя.

Адаптация пользовательских интерфейсов включает следующие аспекты: изменение режима работы в зависимости от функционального состояния пользователя; тестирование как набор тестов, определяющих степень подготовленности пользователя работать с системой, воспринимать предлагаемую информацию; ручная адаптация (пользователь посредством меню или других средств сам выстраивает под себя пользовательский интерфейс, настраивает позиции и порядок пунктов меню, цветовые схемы и т.д.) и другие.

## Методика создания адаптивного интерфейса

Предлагается схема интерфейса, которая предполагает декомпозицию на элементарные приложения (одна адаптивная форма – фрагмент), последующее объединение элементарных приложений в алгоритмическую структуру (структурный элемент) и иерархическое представление структурных элементов с возможностью манипуляции доступа к каждому фрагменту [3].

Первичным при формировании адаптивного интерактивного пользовательского интерфейса является конструирование формы, содержащей различные программные компоненты. Предлагается аппарат конструирования интерфейса на основе автоматной схемы разбора, который включает:

- графический редактор формирования структуры интерфейса;
- формирование схемы разбора интерактива пользователя;
- параметризация интерактивного взаимодействия.

Функции графического редактора используются для создания графического образа интерфейса с интерактивными полями, где:

$G=\{G_i\}_{i=1..l}$  – множество неактивных графических образов (картинка, формула, текст, линия, стрелка перехода, элементы управления и т.п.):

$$V = \{V_j\}_{j=1..J} - \text{множество полей ввода};$$
$$Z = \{Z_n\}_{n=1..N}$$
 – множество полей захвата;

$A=\{A_m\}_{m=1..M}$  – множество альтернативных вариантов интерактива пользователя.

Для формирования схемы анализа интерактива пользователя вводятся понятия позиций, которые связаны с множествами  $Z$  и  $A$ , а именно  $P_Z$  и  $P_A$  как множество действий  $D$ , которые включают:

$D_1 = \langle A_i \rightarrow P_{Z_i} \rangle$  – перетаскивание поля альтернативы;

$D_2 = \langle A_i \rightarrow P_{Ai} \rangle$  – изменение положения альтернативы;

$$D_3 \Leftrightarrow V_{Ai} - \text{ввод текста};$$

$D_4 = \langle OK_i \rangle$  – подтверждение завершения интерактива.

В общем случае формализованное описание интерактива пользователя с возможностью динамической манипуляции присутствием либо отсутствием объектов может быть представлена в виде конечного автомата. Правильность действий формализуется на основе задания конечного автомата, пример которого представлен на рис. 1.

В случае жесткой последовательности он представляет вырожденный случай, где дуги помечаются либо некоторым действием ( $\langle A_i \rightarrow P_{Zi}, \leftrightarrow V_{Ai} \rangle$ ), либо его отрицанием ( $\neg \langle A_i \rightarrow P_{Zi}, \neg \leftrightarrow V_{Ai} \rangle$ ). При таком подходе появляется возможность манипуляции каждым графическим образом пользовательской формы элементарного приложения.

Принято, что любой элемент пользовательского интерфейса может как присутствовать, так и отсутствовать в зависимости от модели пользователя и текущего состояния сценария диалога.

## Формирование алгоритмической структуры пользовательских приложений

Следующим инструментальным средством, обеспечивающим адаптивность пользовательского интерфейса, являются конструкторы пользовательского сценария, которые определяют иерархию приложений, их алгоритмическую структуру и параметризацию с возможностью запуска произвольного количества параллельных процессов [2].

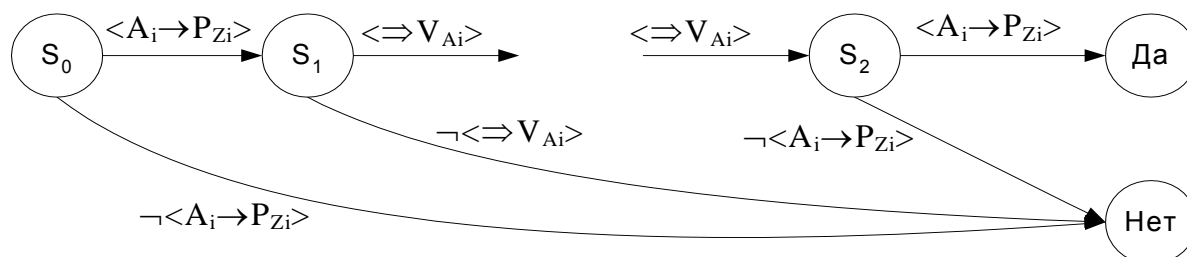


Рис. 1. Пример автоматной схемы формирования интерактива пользователя

Предполагается, что каждому пользовательскому процессу соответствует один инициатор. Если в системе параллельно развивается  $m$  процессов, то в модели присутствует  $m$  инициаторов (особенно актуально для организации синхронизации многопользовательского режима). При наличии эквивалентных параметризуемых приложений, а также реализации механизмов условного перехода по завершению приложения, пользовательский сценарий должен представлять алгоритмическую структуру, которая может быть определена как свертка трека по отношению эквивалентности элементарных операторов пользовательского приложения (рис. 2).

Алгоритмическая структура приложений формализована в виде сетей Петри, которые представляют собой двудольный граф позиций и переходов [3]. Переход имеет множество входов и множество выходов и позволяет моделировать любое логическое условие, что дает возможность создавать механизмы блокировок по входам и любые логические условия перехода по выходам для организации динамического формирования интерфейса.

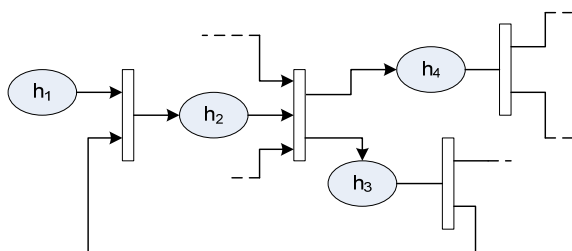


Рис. 2. Алгоритмическая структура приложений

Уровень доступа к фрагменту позволяет для различных групп пользователей автоматически формировать индивидуальные сценарии в зависимости от уровня его подготовки и решаемого круга задач с целью детализации информации (рис. 3).

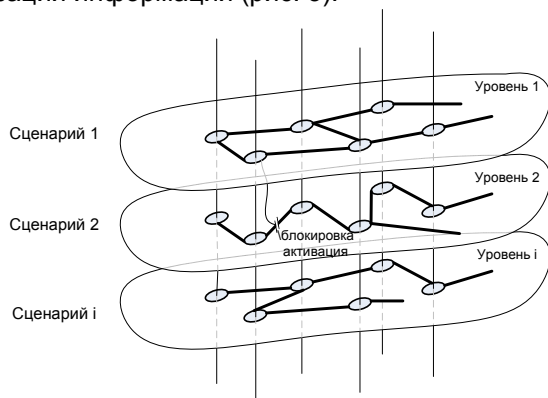


Рис. 3. Структура вложенных сценариев

Кроме того, средства синхронизации позволяют пользователю автоматически переходить на различные уровни сценария взаимодействия.

Для параллельных пользовательских процессов, которые активируются в сетевой среде, реализованы механизмы синхронизации процессов взаимодействия пользователей между собой. Они выполнены на основе общего поля данных по принципу «положил-взял» и непосредственного взаимодействия фрагментов, где  $i$  и  $j$  — фрагменты параллельных ветвей одного сценария (рис. 4).

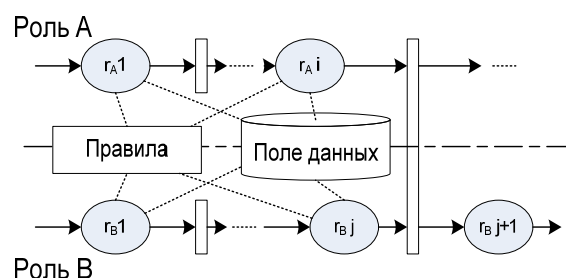


Рис. 4. Механизмы синхронизации и обмена данными

В точках их активации выполняется синхронизация.  $j$  — находится в ожидании, пока фрагмент  $i$  не выложит в общее поле данных ожидаемый информационный ресурс  $r_{ii}$ , либо пока фрагмент  $i$  не передаст фрагменту  $j$  ожидаемые данные непосредственно.  $i$  — по завершению действий пользователя или окончанию обработки каких-либо данных выполняет действие, ожидаемое фрагментом  $j$ . В общем случае, результат выполнения фрагмента  $i$  может влиять на ход выполнения сценария как в роли А, так и в роли В, что и обеспечивает динамику и адаптацию сценария в зависимости от интерактива многопользовательской среды.

#### Задача оптимизации временных характеристик доступа к функционалу

Следующая задача направлена на поддержку создания интерфейса, обеспечивающего оптимальные временные характеристики реализации пользовательских запросов за счет динамической кластеризации ресурсов. Наиболее адекватной формализацией такого процесса являются Марковские цепи (рис. 5), которые учитывают как начальное распределение вероятностей обращения к ресурсам системы, так и вероятности переходов между состояниями, которые определяют распределение ресурсов при реализации пользовательского интерфейса.

Обозначим Марковскую цепь (МЦ) запросов на ресурсы:

$$\xi = (\xi_k, P, P),$$

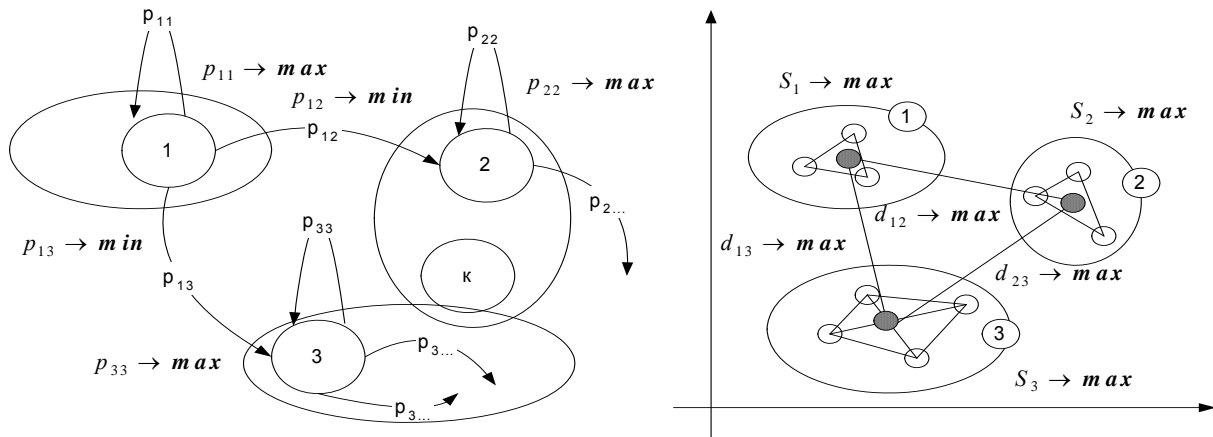


Рис. 5. Преобразование Марковской цепи запросов

где  $\xi_k$  - случайная величина, имеющая конечное число значений, иначе множество состояний  $\mathbf{C}=(\mathbf{C}_0, \mathbf{C}_1, \dots, \mathbf{C}_n)$ ;  $\text{card } \mathbf{C}=n$ ;  $\Pi$  - вектор начального распределения состояний  $\Pi=(p_1, p_2, \dots, p_n)$ ;  $\mathbf{P}$  - матрица переходных вероятностей  $\mathbf{P}=\|p_{ij}\|$   $i, j=1 \dots n$ .

Далее будем рассматривать лишь Марковские цепи, у которых существуют предельные (стационарные) распределения вероятностей  $\pi=(\pi_0, \pi_1, \dots, \pi_n)$ . В данном случае под состоянием ( $\mathbf{C}$ ) цепи будем понимать факт наличия ресурсов у пользователя системы. Вероятности переходов определяются из вероятностей, определяющих последовательность передачи ресурсов каждому пользователю в соответствии с функциональными потребностями.

Для решения этой задачи введены формальные операторы укрупнения, исключения петель, сокращения и другие, которые позволяют построить переходные вероятности вторичной (межгрупповой) цепи по известным характеристикам первичной цепи.

Оператор **укрупнения**  $F_{\text{укр}}^{(C)}$ :  $(\mathbf{C}, \Pi, \mathbf{P}) \rightarrow (\xi', \Pi', \mathbf{P}')$  задается следующим образом. Определяется новое множество состояний и задается отображение  $F_{\text{укр}}^{(C)}: \mathbf{C} \rightarrow \mathbf{C}'$   $\text{card } \mathbf{C}' < \text{card } \mathbf{C}$ . Считается, что если случайная величина  $\xi$  в МЦ  $(\xi, \Pi, \mathbf{P})$  принимает одно из состояний  $(\mathbf{C}_i')^{-1}$ , то в МЦ  $(\xi', \Pi', \mathbf{P}')$   $\xi$  принимает значение  $\mathbf{C}_i'$ . При этом начальное и стационарное распределение вероятностей преобразуются на основании:

$$\Pi' = (p'_1, p'_2, \dots, p'_n),$$

$$\Pi' = F_{\text{укр}}^{(\Pi)}(\Pi),$$

где  $p'_j = \sum_{i \in (\mathbf{C}_j')^{-1}} p_i$ , т.е. суммируются вероятности состояний исходной МЦ, для которой  $F_{\text{укр}}^{(C)}: \mathbf{C}_i = \mathbf{C}_j'$ . Матрица переходных вероятностей

переопределяется оператором  $F_{\text{укр}}^{(P)}$ , как  $\mathbf{P} = F_{\text{укр}}^{(P)}(\mathbf{P}')$ , где:

$$p'_{il} = \sum_{i \in (\mathbf{C}_l')^{-1}} \left( \frac{\pi_i}{\sum_{k \in (\mathbf{C}_l')^{-1}} \pi_k} \sum_{j \in (\mathbf{C}_l')^{-1}} p_{ij} \right).$$

Оператор **исключения петель** представляет  $F_{\text{ип}}^{(P)}: \mathbf{P} \rightarrow \mathbf{P}'$ , где новая МЦ  $(\mathbf{C}', \Pi', \mathbf{P}')$  получается из МЦ  $(\mathbf{C}, \Pi, \mathbf{P})$ , если последнюю рассматривать лишь в моменты перехода из одного состояния в другое. При этом моменты, когда цепь находится в одном и том же состоянии, исключаются. В этом случае  $\text{card } \mathbf{C} = \text{card } \mathbf{C}'$  и  $\forall i p'_{ii} = 0$ .

**Композиция двух предыдущих операций приводит к операции последовательного укрупнения и исключения петель.**

При реализации операции **сокращения** Марковской цепи  $(\mathbf{C}, \Pi, \mathbf{P})$ , множество состояний  $\mathbf{C}$  разбивается на два непересекающихся подмножества  $\mathbf{C}^{(1)}$  и  $\mathbf{C}^{(2)}$ ,  $\mathbf{C} = \mathbf{C}^{(1)} \cup \mathbf{C}^{(2)}$ ,  $\mathbf{C}^{(1)} \cap \mathbf{C}^{(2)} = \emptyset$ . Между  $\mathbf{C}^{(1)}$  и  $\mathbf{C}^{(2)}$  существует биективное отображение  $F_{\text{СМЦ}}^{(C)}: \mathbf{C} \rightarrow \mathbf{C}'$ . Матрица  $\mathbf{P}$  преобразуется следующим образом:

$$F_{\text{СМЦ}}^{(P)}: \mathbf{P} \rightarrow \mathbf{P}',$$

$$p'_{ij} = p_{ij} + p^*_{ij},$$

где  $p^*_{ij}$  определяет вероятность того, что МЦ выйдет из состояния  $i \in \mathbf{C}^{(1)}$  в одно из состояний  $k \in \mathbf{C}^{(2)}$ , и при этом первый выход из множества состояний  $i \in \mathbf{C}^{(2)}$  будет в состояние  $j \in \mathbf{C}^{(1)}$ .

Таким образом, решенная задача позволяет для любой кластеризации состояний определить переходные вероятности между классами состояний, что дает оценку частоты обращения к тем или иным ресурсам и позволяет минимизировать временные затраты на реализацию запросов при динамической настройке пользовательского интерфейса.

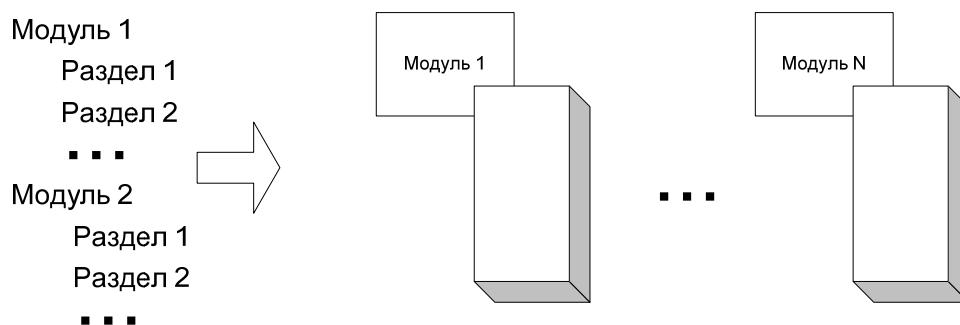


Рис. 6. Формирование адаптивного пользовательского меню

На основе разработанного функционала инструментальных средств формирования адаптивных интерфейсов предлагается динамическая корректировка кластеризации ресурсов с целью минимизации времени реализации пользовательских запросов. Имея статистику по запросам на определенные ресурсы, а также частоту и правильность формирования последовательности функционала каждой группой пользователей, инструментальные средства позволяют динамически настраивать сценарий, открывая только тот набор функций, который необходим конкретному пользователю для решения конкретной задачи.

#### Программные аспекты реализации адаптивных интерфейсов

Рассматриваются вопросы построения программного комплекса формирования адаптивных интерфейсов. Реализована иерархическая структуризация всего доступного функционала. Предлагается динамическое формирование мультимедийного меню, которое реализуется в динамике в зависимости от решаемых задач (рис. 6).

Кроме того, настройки системы позволяют динамически редактировать БД решаемых задач и формируемых динамических фрагментов реализации процедур аналитической обработки.

При проектировании системы использовалась фреймовая технология. Фрейм является основным строительным блоком при сборке конечного приложения. Он инкапсулирует: набор действий, необходимых для решения данной подзадачи; средства для работы и саму логику работы с базой данных (если это предусматривается подзадачей); пользовательский интерфейс, необходимый для диалога пользователя с системой при решении данной подзадачи.

Один и тот же фрейм может быть включен в среды разных категорий пользователей. Технология фреймов обеспечивает встраиваемость одних фрагментов пользова-

тельского интерфейса в другие. Данный подход объединяет преимущества объектно-ориентированной и структурной парадигмы проектирования программных комплексов. При этом объектно-ориентированный подход применяется при проектировании отдельных фреймов, а структурный – при сборке из фреймов конечных приложений.

Именно эта технология предоставляет возможность осуществлять оперативную сборку отторгаемого приложения после отработки сценария в гибридной среде под конкретные требования заказчика.

#### Заключение

В результате разработаны формальные методы и модели описания фрагмента интегрированной пользовательской среды, позволяющие на основе автоматной схемы разбора реализовать методику манипуляции интерактивными графическими объектами пользовательского интерфейса. Разработана методика формирования динамической структуры вложенных пользовательских приложений с учетом уровней доступа пользователя и сложности функционала. На основе формальной модели Марковской цепи запросов на ресурсы, обеспечивающей пользовательский функционал, разработаны модели оценки эффективности распределения ресурсов многопользовательских приложений и механизмы кластеризации ресурсов.

#### Список литературы

1. Срагович, В.Г. Адаптивное управление / В.Г. Срагович. – М.: Наука, 1981. – 384 с.
2. Бенгеддаш, С. Общие принципы формирования адаптивного WIMPS-интерфейса. / С. Бенгеддаш // Моделирование технолог. процессов в пром-ти и образовании: сб науч. тр. / МАДИ(ГТУ). – М., 2004. – С. 138-145.
3. Суворов, Д.Н. Разработка двухсторонних адаптивных интерфейсов / Д.Н. Суворов, С. Бенгеддаш, С.В. Мазуренко // Моделирование технолог. процессов в пром-ти и образовании: сб науч. тр. / МАДИ(ГТУ). – М., 2004. – С. 174-177.

**Methods and models of formation of adaptive user interface systems in support of administrative decisions**

E. Moskvichev, P. Timofeev, S. Satyshev,  
A. Solntsev

**Abstract.** The article deals with the topic of development of the instrumental tools and methods for design of the adaptive user-interfaces. The device for interface design based on automatic circuit analysis is considered by the authors. The also article implies requirements for graphic editor to design interface structures and schemes for user circuits breakdown. The algorithmic structure of the applications is

formalized in a form of Petri nets. The article provides analysis of the temporal characteristics optimization for implementation of user requests through dynamic clustering of the resources. Authors of the article suggest formalization in the form of Markov's chain which calculates the initial probability distribution of access to system resources.

**Keywords:** interface, adaptation, script, nested processes.

Рецензент: П.Ф. Юрчик, д-р техн. наук, проф. МАДИ

Статья поступила 01.11.2011

УДК 005.511

**В.Н. Брыль**

**ОЦЕНКА ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ  
БИЗНЕС-ПРОЦЕССАМИ**

**Аннотация.** В статье рассматриваются факторы, определяющие эффективность управления бизнес-процессами. На основании анализа этих факторов проводится экономико-математическое моделирование эффективности бизнес-процесса управления с целью разработки интегрального показателя эффективности.

**Ключевые слова:** бизнес-процесс, оценка эффективности управления, стоимость управленческих решений, оперативность принятия управленческих решений, международные стандарты.

**Введение**

Современная сложная и динамичная рыночная среда требует от предприятий постоянного совершенствования своих систем управления и информационных систем их поддержки. Сегодня общепризнанным является факт, что кризис экономики - это, прежде всего, кризис управления, суть которого состоит в несоответствии качества управления предприятиями требованиям современной рыночной среды. Выход из такого положения известен – необходимо внедрение эффективных механизмов управления в соответствии общепринятыми мировыми стандартами. В частности, такими стандартами являются международные стандарты ISO 9001 и 9004 «Система Менеджмента Качества».

**Управление процессами в соответствии со стандартами ISO 9001, 9004**

В соответствии со стандартом ISO 9001 к управлению бизнес-процессами предъявляются следующие требования:

- определение критериев и методов, необходимых для обеспечения результативности

как при осуществлении, так и при управлении этими процессами;

- принятие мер, необходимых для достижения запланированных результатов и постоянного улучшения этих процессов. Организация должна осуществлять менеджмент бизнес-процессов в соответствии с требованиями настоящего стандарта.

В стандарте ISO 9004 к управлению бизнес-процессами выделены следующие требования:

- определение систем и процессов, которые могут быть четко поняты, подвергнуты менеджменту и улучшены с точки зрения результативности и эффективности;
- обеспечение результативного и эффективного выполнения и управления бизнес-процессами, а также показателями и данными для определения удовлетворительной деятельности организации.

Основная мысль этих требований заключается в том, что недостаточно формализовать бизнес-процессы предприятия, необходимо еще и обеспечить эффективное управление ими.