

УДК 69

DOI 10.21661/r-119315

Т.А. Мурзажанов, Е.Б. Степанова, Н.В. Максимов

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ОТЧЕТНОСТИ ДЛЯ БАНКОВСКОЙ ОТРАСЛИ

Аннотация: в данной статье рассматривается математический подход к созданию интерфейса системы отчетности на примере банковской отрасли с целью управления и улучшения работы банка. На основе подхода разработана система отчетности с последующим ее внедрением в банке. Данная система отчетности предназначена для поддержки и повышения эффективности проведения совещаний руководства, что, в свою очередь, повышает эффективность работы банка.

Ключевые слова: система отчетности, проектирование интерфейсов, минимаксная задача, ключевые показатели эффективности, автоматное программирование.

T.A. Murzazhanov, E.B. Stepanova, N.V. Maksimov

BANKING AREA REPORTING SYSTEM DESIGN

Abstract: the article concerns with the mathematical approach to the creation of the interface reporting system on the sample of the banking area in order to manage and improve the banking performance. It is based on the approach developed reporting system, with the following implementation in the Banking structure. This reporting system is designed to support and enhance the effectiveness of leadership meetings, which increases the efficiency of the bank.

Keywords: reporting system, interface design, minimax, key performance indicators, automata-based programming.

Для эффективного управления предприятием необходимо разработать систему отчетности для поддержки и повышения эффективности проведения совещаний руководства, что, в свою очередь, повышает эффективность работы предприятия. Подобный подход позволяет сократить степень вмешательства человека в подготовку данных путем автоматизации сбора информации. Кроме того, подобная система сопряжена с созданием единого стандарта данных для предприятия. Единая система и методология отчетности перед руководством приводит к улучшению как понимания самого процесса ведения совещаний, так и процессов внутри предприятия. Отчетность проводится в установленное время на совещании, что делает ее менее субъективной.

Основная цель системы отчетности заключается в поддержке и повышении эффективности проведения совещаний руководства, что, в свою очередь, повышает эффективность работы предприятия. Приложение включает в себя адаптивное представление ключевых показателей эффективности каждого департамента, что позволяет выявить слабые стороны, исходя из ключевых показателей эффективности (КПЭ).

Пусть перед оператором поставлена задача, которую он может выполнить после проведения определенных действий в пользовательском интерфейсе приложения. Представим решение задачи с ограничениями в виде системы следующего вида [1, с. 109]:

$$\begin{cases} \min \sum_{i=1}^m f(x_i, X') \geq n \\ \max_{Y \in G} F(X', Y) \rightarrow \min_{X' \in \Omega} \end{cases}$$

где $X' = ((x_1, c_1), \dots, (x_n, c_n))$ – мультимножество элементов интерфейса и функций x_i , из которых состоит исходный интерфейс, c_i – количество вхождений элементов и функций в интерфейс, а $Y = (y_1, \dots, y_n)$ – множество подзадач, которые следует выполнить оператору. В свою очередь Ω – замкнутое мультимножество, а G – ограниченное замкнутое множество.

Рассмотрим ограничения, которые накладывает минимаксная задача. Начнем с множества G , которое является ограниченным замкнутым множеством.

В самом деле, если мы рассмотрим множество подзадач Y , которые нужно решать оператору, то мы увидим, что множество G является ограниченным исходя из определения ограниченного множества. Так как найдется $M > 0$ такое, что $|y_i| \leq M, \forall y_i \in Y, Y \subset G$. Пусть множество G и мультимножество Ω заданы таким образом, что для них определена операция объединения, т.е. $\exists A \subset G$ такое, что $A = B \cup C$, где $B \subset G$ и $C \subset G$. Аналогично с мультимножеством Ω . Замкнутость множества обусловлена тем, что любой элемент из множества G можно получить из комбинации других элементов того же самого множества, путем операции объединения. В самом деле подзадачи могут решаться с помощью комбинации других подзадач. Замкнутость множества Ω обусловлена теми же условиями, что и замкнутость множества G . Любые элементы и функции интерфейса g могут быть реализованы через комбинацию других элементов этого множества.

Рассмотрим условия работы с интерфейсом, что позволит выделить подмножества X' и Y из мультимножества Ω и множества G соответственно. Данный интерфейс будет использоваться на совещаниях. Оно будет создано под планшетное устройство. На совещании будет присутствовать около десяти участников (операторов). У каждого оператора будет по одной машине (планшету). Помимо ведения живого диалога между присутствующими, может возникнуть потребность в рассмотрении конкретной проблемы, например, не выполнение плана одним из отделений департамента [2, с. 460]. У каждого оператора должен быть доступ к данному ключевому показателю эффективности, чтобы наглядно увидеть проблему на планшете. Исходя из регламентированного времени проведения совещания в приложение должен быть встроен таймер, который позволял бы грамотно распределить время. У присутствующих должна быть возможность проводить отчетность непосредственно на совещании, следовательно, следует включить в программу возможность просмотра презентаций.

Перейдем к определению матрицы стратегий из множества подзадач и мультимножества функционала. Определим максимальный элемент множества Y по

столбцам: $Y^* = \max_j y_j$ и минимальный на комплекте X' по строкам: $X'^* = \min_i x'_i$. Данная точка и будет являться седловой точкой и решением выше поставленной минимаксной задачи.

Приступим к описанию элементов пользовательского интерфейса с точки зрения программирования. Для описания элементов удобно и естественно использовать объектный подход. При этом имеет смысл выделить базовый класс для всех элементов пользовательского интерфейса. При разработке автоматного подхода к реализации элементов пользовательского интерфейса мы также будем использовать принципы объектно-ориентированного программирования.

В целях демонстрации предлагаемого подхода была спроектирована система классов, реализующих несколько широко применяемых элементов пользовательского интерфейса и средства их взаимодействия с внешним относительно системы автоматов миром. Поскольку, целью данного примера является демонстрация автоматного подхода, то мы постараемся ограничиться базовой функциональностью элементов управления. Каждый элемент пользовательского интерфейса реализован в виде класса. Классы элементов пользовательского интерфейса разделяются на элементы управления (<Bubble/>, <StartButton/>, <Switcher/> и т. д.) и контейнеры (<MainView/>, <BubbleView/>, <CalendarView/> и т. д.).

Обработка событий в автомате производится в методе `changeState(event)`, принимающем в аргументах идентификатор события. Каждый класс в иерархии наследования является отдельным автоматом, имеет свое собственное состояние и свой собственный граф переходов. Метод `changeState` реализован во всех классах. Поэтому событие, в первую очередь, обрабатывается конечным классом в иерархии наследования. При необходимости класс может передать полученное сообщение своему непосредственному родителю или вызвать автоматный метод родителя с любым другим сообщением. События поступают в систему автоматов через класс сцены, преобразующий сообщения операционной системы в события для автоматов. Объект класса сцены в свою очередь с использованием механизма

класса конкретного контейнера (View) передает сообщение содержащимся в нем элементам интерфейса верхнего уровня, для которых это сообщение актуально (например, при выборе активного элемента событие должно быть передано следующему (открывающемуся по нажатию) окну и окну, которое было активно в предыдущий момент времени). Те из них, которые являются контейнерами, передают событие ниже по иерархии принадлежности объектам, для которых это событие актуально. Стоит отметить, что при работе с приложением, при запуске любой функции, которая влияет на изменение элементов интерфейса, запускается функция `changeState()`, которая определяет, какой элемент интерфейса изменил свое состояние и меняет его состояние. То есть количество взаимодействия оператора с интерфейсом равно количеству запуска функции `changeState()`. Минимизация количества запусков функции `changeState()` и есть приведение задачи к нахождению седловой точки и оптимизации работы приложения.

После создания системы отчетности была проведена оценка полученного пользовательского интерфейса гибридным методом GOMS. Метод основан на минимизации выполнения действий пользователя. Применение метода показало, что оператор тратит более чем на треть меньше времени при использовании визуальной структуры системы отчетности, построенной с учетом минимаксной задачи. Полученное решение было успешно внедрено на предприятии, заменив предыдущую систему проведения совещаний и ведения отчетности.

Список литературы

1. Мурзажанов Т.А. Проектирование функционально-ориентированных интерфейсов / Т.А. Мурзажанов // Стратегии развития науки и образования в XXI веке: Сборник научных трудов по материалам Международной науч.-практ. конф. (30 ноября 2016 г.). – Смоленск: Новаленсо, 2016. – С. 108–109.
2. Степанова Е.Б. Разработка и использование методов описания процессов и документопотоков для систем терминального управления документооборотом / Е.Б. Степанова, А.В. Тимофеев, Н.В. Максимов // Аудит и финансовый анализ. – 2012. – №4 – С. 457–462.

Мурзажанов Тимур Артурович – магистрант ФГАОУ ВО «Национальный исследовательский ядерный университет «МИФИ», Россия, Москва.

Murzazhanov Timur Arturovich – graduate student of FSAEI of HE “National Research Nuclear University MEPhI”, Russia, Moscow.

Степанова Елена Борисовна – канд. физ.-мат. наук, старший научный сотрудник, доцент, руководитель группы «Интегрированные системы» ФГАОУ ВО «Национальный исследовательский университет «МИФИ», Россия, Москва.

Stepanova Elena Borisovna – candidate of physico-mathematical sciences, senior research officer, associate professor, the head of the «Integrated system» of FSAEI of HE “National Research Nuclear University MEPhI”, Russia, Moscow.

Максимов Николай Вениаминович – д-р техн. наук, профессор кафедры системного анализа «Национальный исследовательский университет «МИФИ», Россия, Москва.

Maksimov Nikolay Veniaminovich – doctor of engineering sciences, professor at the department of System analysis of FSAEI of HE “National Research Nuclear University MEPhI”, Russia, Moscow.
