



Об авторах

Виктор Ефимович Пониматкин — канд. техн. наук, доц., Балтийский федеральный университет им. И. Канга, Калининград.

E-mail: VPonimatkin@kantiana.ru

Андрей Алексеевич Шпилевой — канд. физ.-мат. наук, доц., Балтийский федеральный университет им. И. Канга, Калининград.

E-mail: AShpilevoi@kantiana.ru

68

About the authors

Viktor Ponimatkin — PhD, Senior Research Fellow, Ass. Prof., I. Kant Baltic Federal University, Kaliningrad.

E-mail: VPonimatkin@kantiana.ru

Andrey Shpilevoy — PhD, Ass. Prof., director of FTI, I. Kant Baltic Federal University, Kaliningrad.

E-mail: AShpilevoi@kantiana.ru

УДК 519.715

А. А. Меркулов

АВТОМАТИЗАЦИЯ СИНТЕЗА СИТУАЦИОННЫХ ЦЕНТРОВ НА ОСНОВЕ УНИВЕРСАЛЬНОГО ПАТТЕРНА ОРГАНИЗАЦИЙ VSM CENOSE

Сегодня существует большая потребность в компьютерных системах поддержки принятия решений класса «ситуационный центр» на федеральном и региональном уровне для крупных, средних и малых предприятий. Предлагается технология автоматизации синтеза ситуационных центров для любых предметных областей, упрощающая проектирование ситуационного центров и снижение его стоимости.

Today there is a great need for computer systems that support decision-making class situation centre at the Federal and regional level, for large, medium and small enterprises. The technology of auto-automation of the synthesis of the situational centers for all subject areas, to simplify the design of situational centers and reducing its cost.

Ключевые слова: компьютерная система поддержки принятия решений, ситуационный центр, технология автоматизации синтеза, гибридный паттерн.

Key words: computer system that support decision-making, situation centre, technology of auto-automation of the synthesis, hybrid pattern.



Введение

Одно из эффективных средств повышения качества принятия коллективных решений — ситуационный центр (СЦ) [1] — компьютерная система поддержки принятия решений (КСППР), построенная на целостном подходе, отображающем все аспекты деятельности системы управления. Разработка (проектирование) СЦ — сложный, наукоемкий и трудоемкий процесс, включающий взаимообусловленный комплекс математических, информационных, программных, технических, организационных, правовых и других обеспечений. Отсюда и высокая стоимость проектирования, по экспертным оценкам составляющая на федеральном уровне миллиарды, а на региональном уровне — десятки и сотни миллионов рублей [2].

В работе А. В. Колесникова и А. А. Меркулова [3] разработана технология синтеза ситуационных центров. В ней показано, что создание СЦ не заканчивается на этапе его внедрения. СЦ должен развиваться в процессе функционирования из-за увеличения/уменьшения размеров организации, изменения предметной деятельности и других параметров. Однако остался открытым вопрос автоматизации данного процесса. И сегодня по мере алгоритмизации процесса создания ситуационного центра очень актуальны новые подходы, обеспечивающие технологию автоматизированного синтеза СЦ и его сопровождения на всех этапах его жизненного цикла.

Постановка задачи

В ситуационном центре можно выделить организационную, техническую и программную подсистему [1]. Проектирование организационной и технической компоненты сегодня достаточно отработано [1; 4] и не требует особых системных исследований и научных усилий. Все решается на инженерном уровне. Основные научные и технические сложности синтеза ситуационного центра и затраты на функционирование и сопровождение определяет программное обеспечение. Особенно та его часть, которая связана с моделированием организации (объекта управления) и воздействий внешней среды. В предыдущих работах автора [3–6] рассмотрена общая схема автоматизации построения модели объекта управления ситуационного центра.

Однако практика показывает, что для эффективного использования ситуационных центров [4] хорошей модели недостаточно. Необходимо обеспечить глубокую взаимосвязь между инструментальными средствами, методологией разработки и вопросами эксплуатации постоянно модернизируемого СЦ [3].

При этом разработчики и потребители становятся как субъектами, так объектами создания, эксплуатации ситуационного центра и его непрерывной модернизации. С одной стороны, создаются схемы, функ-



ции и алгоритмы уникального СЦ, а с другой — нарабатывается ядро стереотипных решений, в разы снижающее финансовые, технологические и временные издержки при производстве ситуационного центра для другого заказчика на основе накопленного опыта [3]. Сформулированные выше требования при одновременном следовании парадигмам компонентной ориентации [7] и экстремального программирования [8] дают существенное упрощение синтеза СЦ [3].

В работе [3] удалось формализовать процессы, которые происходят в плоскости взаимодействия концептуальных моделей и инструментальных средств. Однако не было алгоритмизации процессов, которые возникают при взаимодействии разработчиков/пользователей и концептуальных моделей, а также разработчиков/пользователей и инструментальных средств.

Таким образом, ускорение и упрощение СЦ можно достичь при учете взаимодействия разработчиков (разработка), концептуальных моделей и инструментальных средств, алгоритмизации и автоматизации этих процессов в едином пространстве.

Анализ существующих технологий

Известно достаточное количество работ по всем составляющим синтеза СЦ на всех этапах его жизненного цикла. В обобщающей работе М. В. Губко и Н. А. Коргина [9] выделяется 8192 класса возможных подходов к анализу и синтезу организационных структур. Авторами в качестве основного метода синтеза организаций выбрана методика синтеза на основе гибридного паттерна VSM Cenose [3–6]. Однако в этих трудах не рассматривались связь разработчиков с инструментальными средствами и разработчиков с концептуальной моделью.

В части разработки программного обеспечения выделяются следующие основные подходы [10]: стихийное программирование, структурный подход, объектный подход, компонентный подход, Case-технологии, блочно-иерархический подход, экстремальное программирование. В последнее время сформировалось понятие жизненного цикла программного обеспечения. Процесс жизненного цикла определяется как совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. На протяжении последних тридцати лет в программировании сменились три модели жизненного цикла программного обеспечения; каскадная, модель с промежуточным контролем и спиральная [10]. В работе [3] показана связь инструментальных средств с моделью, но нет связи с разработчиками/пользователями и их терминологической базой.

Для разработчиков/пользователей существуют справочники, систематизирующие термины и определения по конкретным предметным областям, технологиям программирования, способам проектирования баз данных и т. д. Особенно можно выделить работу А. М. Новикова и Д. А. Новикова по созданию словаря основных понятий общесистемной методологии [11]. Однако объединяющей методики для разработки, концептуальных моделей и инструментальных средств пока не существует.



Технология синтеза ситуационного центра

В рамках предлагаемой технологии требуется обеспечить глубокую взаимосвязь между концептуальной моделью, инструментальными средствами и разработкой. Это обеспечит ускорение внедрения, упрощение функционирования и модернизации ситуационного центра. Разработчики и пользователи становятся как субъектами, так объектами управления в процессе жизненного цикла СЦ. При этом, с одной стороны, создаются схемы, функции и алгоритмы уникального СЦ, а с другой – нарабатывается ядро стереотипных решений, в разы снижающее финансовые, технологические и временные издержки при производстве СЦ для другого заказчика на основе накопленного опыта.

В качестве концептуальной модели используется паттерн VSM Cenose [5]. Паттерн строится на идеях технетики [12], организационной кибернетики [13] и аутопойезиса [14; 15]. Он описывает ресурсные, структурные и инвариантные свойства организации (рис. 1).

Аутопойезис	Технетика	Модель VSM
<ul style="list-style-type: none"> - Целостность - Граница - Редукция - Комплексности - Контингенция - Комплексности - Операционная за- - мкнутость - Самореференция - Коммуникация 	$\left\{ \begin{aligned} \sum_{j=1}^{\infty} \left(\int_0^{\infty} W_j(x) dx \right) &= \int_0^{\infty} \Omega(y) dy \cdot \sum_{j=1}^{\infty} \left(\int_{t_{j1}}^{t_{j+1}} W_j(x) dx \right) = W_{\Sigma} = \text{const}; \\ \int_0^{\infty} \Omega(y) dy &= \max; \\ \int_0^{\infty} W_j(x) dx &= \int_{j_{\max}}^{\infty} W_j(x) dx; \\ \int_{t_{j1}}^{t_{j+1}} W_j(x) dx &= \Lambda(t_{j1}) \cdot M[W_j(t_{j1})] = W_{j\Sigma} = \text{const}; \\ t_{j1} &= \int_{t_{j1}}^{\infty} \Lambda(x) dx; \\ \sum_{j=1}^{\infty} \left(\int_{t_{j1}}^{t_{j+1}} \omega_j(x) dx \right) &= \sum_{j=1}^{\infty} \left(\int_{t_{j1}}^{t_{j+1}} \mu_j(x) dx \right) = \frac{W_{j\Sigma}}{2} = \text{const}; \\ \sum_{j=1}^{\infty} \left(\int_{t_{j1}}^{\infty} \omega_j(x) dx - \int_0^{\infty} \mu_j(x) dx \right) &= 0; \\ \sum_{j=1}^{\infty} \left(\int_0^{\infty} \omega_j(x) dx + \int_0^{\infty} \mu_j(x) dx \right) &= \sum_{j=1}^{\infty} \left(\int_0^{\infty} W_j(x) dx \right). \end{aligned} \right.$	
Не рассмотрены свойства		
Структурные, ресурсные	Структурные, инвариантные	Ресурсные, инвариантные
Паттерн VSM Cenose реализует в одном конструкте ресурсные, структурные и инвариантные свойства, самовоспроизводящиеся в организации		

Рис. 1. Паттерн VSM Cenose

Для связи паттерна VSM Cenose с инструментальным уровнем формируется метамодель на основе модели хранения данных «Категории сущностей и связей» (КСС) [16]. И далее уже строятся конкретные и уникальные модели.

Направление движения от метамодели к уникальной реализации назовем концептуальным направлением. Двигаясь таким образом мы проходим различные модели предметного мира: метамодель (КСС), абстрактная модель (паттерн VSM Cenose), конкретная модель (типовое



проектное решение – ТПП) и уникальная модель (реализация). Мета-модель задает структуру, а реализация – конкретный предмет, то есть мы движемся от структуры к предмету (рис. 2).



Рис. 2. Концептуальное направление

Для использования инструментальных средств будем применять технологию компонентно-ориентированного [7] и экстремального [8] программирования на всех стадиях разработки, внедрения, функционирования и модернизации СЦ. Эти технологии необходимо связать с концептуальным направлением и с разработчиками.

Компонентно-ориентированное программирование подразумевает [7]: объектно ориентированное программирование, модульность (включая инкапсуляцию информации и позднее связывание модулей), безопасность (статический контроль типов переменных и автоматическое управление памятью), наследование реализации через границы модулей.

Экстремальное программирование подразумевает [8]: короткий цикл обратной связи (разработка через тестирование, заказчик всегда рядом), непрерывный процесс (непрерывная интеграция, рефакторинг, частые небольшие релизы), понимание разделяемое всеми (простота, метафора системы, коллективное владение кодом, стандарты кодирования).

Введем понятие абстрактности и конкретности инструментальных средств и производимых по отношению к разработчику/пользователю. Наиболее абстрактная часть – база данных. Пользователь «не видит» и «не знает», как база работает, а конкретная – контент, это то, что всегда перед глазами. Таким образом, инструментальные средства проектирования ситуационного центра делятся на СУБД, средний слой, интерфейс, контент. Это движение обозначает движение от абстрактного к конкретному (рис. 3).

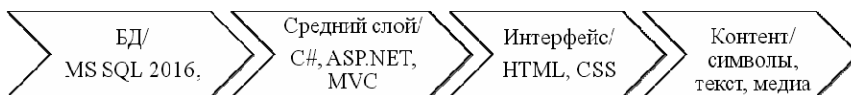


Рис. 3. Инструментальное направление

Опишем направление разработки. В этом процессе участвуют различные группы специалистов. Мы выделим следующие ключевые роли: архитекторы, разработчики, прикладные разработчики, тестировщики. На разных этапах к ним присоединяются менеджеры, внедренцы и пользователи. Из множества параметров, которыми можно описать специалистов, в качестве критерия употребим понятийный аппарат, который использует та иная группа.



В целом весь понятийный аппарат должен быть, с одной стороны, понятен всем, а с другой — иметь различные слои, которые неразрывно связаны не только друг с другом, но и с концептуальным и инструментальным направлением.

В части систематизации терминов и определений известны справочники по конкретным предметным областям, технологиям программирования, способам проектирования баз данных и т. д. Можно отметить работу А. М. Новикова и Д. А. Новикова по созданию словаря основных понятий общесистемной методологии [11].

Однако на сегодня нет словарей и справочников, которые бы обеспечивали взаимосвязь понятий различных системных уровней для конкретно разрабатываемой системы поддержки принятия решений в конкретной организации.

В рамках предлагаемой технологии созданы понятийный аппарат:

- 1) для метамодели — язык архитекторов системы;
- 2) для абстрактной модели — язык разработчиков.

Далее идут понятия конкретной модели, которые определяются языком спецификаций предметной области. Этот аппарат в основном используется прикладными программистами.

Понятия уникальной модели связаны с пользователями и тестировщиками, которые принимают окончательный вид в процессе заполнения системы данными по объекту.

Таким образом, направление разработки можно разбить на четыре участка: архитектурный, участок разработки, прикладного программирования, тестовый (рис. 4).

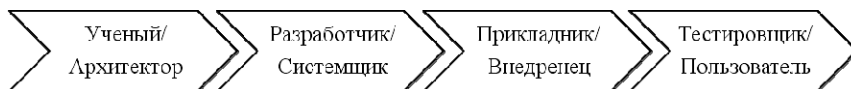


Рис. 4. Направление разработки

При этом все работы по трем направлениям должны быть решены одновременно на всех стадиях жизненного цикла ситуационного центра. В этом случае тезаурус терминов и определений, опираясь на модель синтеза организаций, обеспечит необходимую связь структурных и абстрактных понятий с конкретным языком пользователя, а экстремальное программирование сделает этот процесс динамичными и уникальным для разработчиков.

Методика синтеза организаций обогатится процессом постоянной верификации в практике и все большим уточнением строения своего ядра и его границ.

Методика экстремального программирования обогатится арсеналом программных и методических инструментов, обеспечивающих синтез ситуационного центра без программирования.

Объединяя все три подхода, мы сможем в рамках жизненного цикла ситуационного центра решать задачу поиска общих закономерностей

(номотетический подход) синтеза ситуационного центра и исследовать ее уникальное состояние (идеографический подход) в процессе внедрения ситуационного центра в конкретную организацию.

В рамках внедрения номотетические и идеографические подходы совмещаются. Номотетический подход дает возможность разработчикам постоянно совершенствовать структурно-абстрактное ядро комплекса, которое генерализует общие закономерности строения организации, а идеографический подход позволяет пользователям в динамике внедрения ситуационного центра сделать его уникальными именно для конкретной организации в данный момент времени. Полученный опыт может тиражироваться как горизонтально, так и вертикально.

74

Отобразим три направления и их деление на четыре в части в виде трехмерной системы координат (рис. 5). На оси абсцисс отражено направление разработки от архитектора системы до тестировщика, где X может принимать 4 значения: X_1, X_2, X_3, X_4 .

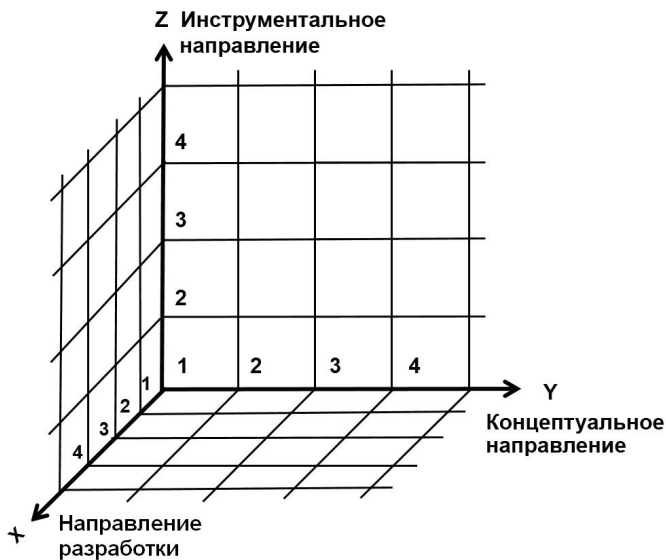


Рис. 5. Схема направлений синтеза

На оси ординат отражено концептуальное направление от структуры к предмету, где Y может принимать 4 значения: Y_1, Y_2, Y_3, Y_4 .

На оси аппликат отражено инструментальное направление от абстрактного к конкретному, где Z может принимать также 4 значения: Z_1, Z_2, Z_3, Z_4 .

Таким образом, мы получаем 64 куба. С одной стороны, каждый куб естественным образом связан с другими через модели, инструментальные средства и процесс проектирования, а с другой — достаточно независим для проведения работ аналитиков и программистов соответствующего уровня подготовки. Технология синтеза ситуационного центра для конкретной организации восходит от структурно-абстрактного уровня, выполняемого архитекторами и учеными, к предметно-конкретному, возникающему в результате деятельности тестировщиков и пользователей конкретной организации.



Для дальнейшей работы дадим формальные определения понятий, которые используются на всех трех направлениях. Определения даны применительно к задачам, решаемым в рамках синтеза ситуационного центра и могут не совпадать с определениями, которые даны для других предметных областей.

Определим термины и определения по направлению абсцисс (X). *Архитектор/ученый* (X_1) – специалист, разрабатывающий метамодели организационных структур и описывающий их на языке, который не зависит от предметной области и инструментальных средств.

Разработчик/системотехник (X_2) – специалист, разрабатывающий абстрактные модели организационных структур и описывающий их на языке, который не зависит от предметной области и инструментальных средств.

Прикладной программист/инженер/внедренец (X_3) – специалист, разрабатывающий конкретные модели и описывающий их на языке, который *зависит* от предметной области и инструментальных средств.

Пользователь/тестировщик (X_4) – специалист, работающий с предоставленными ему данными и инструментальными средствами, который тестирует и эксплуатирует систему.

Определим понятия по направлению ординат (Y). *Метамодель* (Y_1) – модель, которая описывает базовые понятия, правила, структуру и принципы действия других более высокоуровневых моделей. В нашем случае это абстрактная и конкретная модель. В роли метамодели выступает модель хранения данных «Категории сущностей и связей» (КСС).

Модель данных КСС (Y_1) – высокоуровневая модель хранения данных, обеспечивающая проектирование предметных областей без перепрограммирования ядра программного комплекса.

Абстрактная модель (Y_2) – модель, отражающая наиболее общие характеристики объекта. В нашем случае это паттерн VSM Cenose, наследующий свойства КСС и обладающий развернутым описанием организационных структур.

Паттерн VSM Cenose (Y_2) – ограниченная в пространстве и времени самовоспроизводящаяся совокупность элементов и кластеров, обладающая VSM подобной структурой управления и ценологическими свойствами [5]. Выступает в роли метамодели для конкретной модели.

Конкретная модель (Y_3) – модель, построенная на базе абстрактной модели, учитывающая специфику предметной области: вопросы бюджетирования для финансовой системы, способы учета для бухгалтерии и т.д. Количество конкретных моделей соответствует числу моделируемых предметных областей.

Типовое проектное решение (Y_3) – комплект технической документации, содержащий проектные решения по части объекта проектирования, включая программные средства, и предназначенный для многократного применения в процессе разработки, внедрения и функционирования АСУ с целью уменьшения трудоемкости разработки, сроков и затрат на создание АСУ и ее частей.



Реализация (Y_4) — совокупность всех используемых моделей применительно к данной предметной области вместе с данными.

Модель организации (Y_4) — модель, построенная на базе конкретной модели предметной области с учетом уникальных структурных особенностей организации.

Определим термины и определения по направлению приложений (Z). СУБД (*система управления базами данных*) (Z_1) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

MS SQL 2014 (Z_1) — система управления реляционными базами данных (СУРБД), разработанная корпорацией *Microsoft*, версия 12.0.2456.0. В качестве базы данных могут также использоваться любые базы данных с кластеризованными индексами (*Oracle*, *MySQL*, с определенными ограничениями *PostgreSQL*) и др.

Средний слой (Z_2) — слой или комплекс технологического программного обеспечения для взаимодействия между различными приложениями, системами и компонентами. В данном комплексе для организации среднего слоя используются плагины, *ASP.Net*, *MVC*, *IIS*.

Плагин (Z_2) — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования ее возможностей.

ASP.NET (англ. *Active Server Pages*) (Z_2) — технология создания веб-приложений и веб-сервисов от компании Майкрософт. Она является составной частью платформы *Microsoft .NET*.

MVC (англ. *Model-view-controller*, «модель-представление-контроллер», «модель-вид-контроллер») (Z_2) — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

IIS (англ. *Internet Information Services*) (Z_2) — набор серверов для нескольких служб Интернета от компании Майкрософт.

Интерфейс (Z_3) — совокупность способов и методов взаимодействия двух систем, устройств или программ для обмена информацией между ними. В качестве инструмента построения интерфейса используется *HTML*, *CSS*, *Javascript*.

HTML (англ. *HyperText Markup Language* — «язык гипертекстовой разметки») (Z_3) — стандартный язык разметки документов во Всемирной паутине.

CSS (англ. *Cascading Style Sheets*, каскадные таблицы стилей) (Z_3) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Javascript (Z_3) – прототипно-ориентированный сценарный язык программирования.

Контент (Z_4) – информационное наполнение сайта, программы: тексты, графическая, аудио-, видеоинформация и т. д.

На рисунке 5 мы видим, что образовались четыре тройки:

1) X_1, Y_1, Z_1 – архитектор, метамодель, база данных (область конструкторов);

2) X_2, Y_2, Z_2 – разработчик, абстрактная модель, средний слой (область паттерна);

3) X_3, Y_3, Z_3 – прикладной программист, конкретная модель, интерфейс (область типовых проектных решений);

4) X_4, Y_4, Z_4 – пользователь, уникальная модель, контент.

Таким образом, задачу синтеза уникального ситуационного центра на этапах создания, сопровождения и модернизации можно разделить на 4 части.

Первая часть. Синтез базы данных на основе метамодели. Это обеспечивается архитекторами системы. В качестве терминов используются понятия модели хранения данных «Категории сущностей и связей» [16]. Разработан специализированный инструмент: база данных [17] и система управления базой данных [18].

Синтез среднего слоя на основе абстрактной модели. Это обеспечивается разработчиками системы. В качестве терминов используются понятия абстрактной модели [5]. Средний слой строится на паттерне VSM Cenose [4] и плагинной архитектуре [19]. Схема организации среднего слоя приведена на рисунке 6.

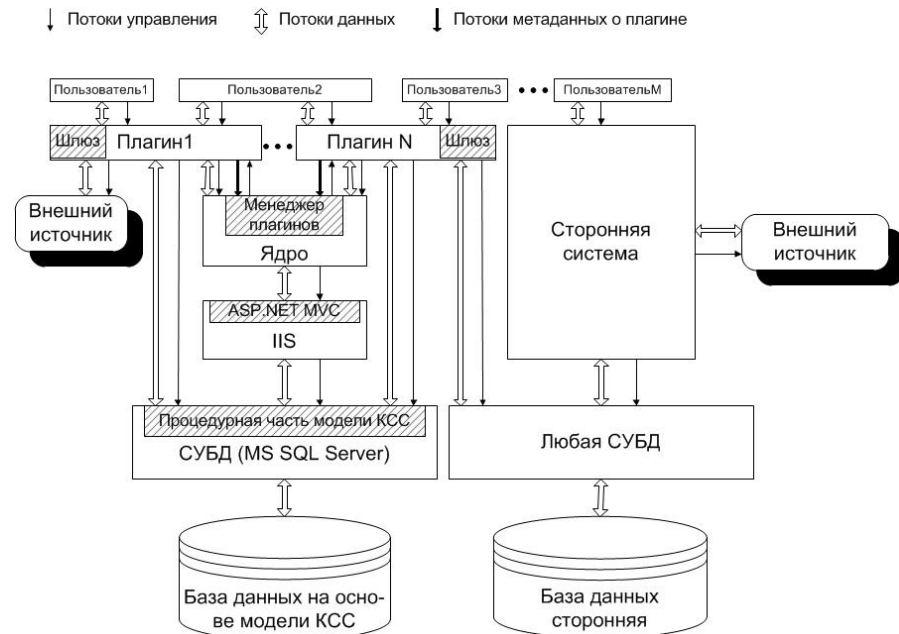


Рис. 6. Схема организации среднего слоя

Синтез интерфейса на основе типового проектного решения [20]. Это обеспечивается прикладными программистами во взаимодействии с пользователями. Основными понятиями формируются из спецификаций конкретной предметной области.

Разработан специальный инструмент [21], топология которого представлена на рисунке 7.



78

Рис. 7. Схема топологии интерфейса

Особенность данного подхода заключается в том, что мы видим картину в целом и детальную информацию совместно. Целостная картина представлена, как правило, графическими образами, а детальная информация — символьными полями.

Совокупность конфигураций образует рабочее место. Взаимосвязь между рабочими местами, конфигурациями и виджетами приведена на рисунке 8. У каждого рабочего места может быть много пользователей. Нижний индекс справа — номер рабочего места. Каждое рабочее место имеет некоторое количество конфигураций KF1M.

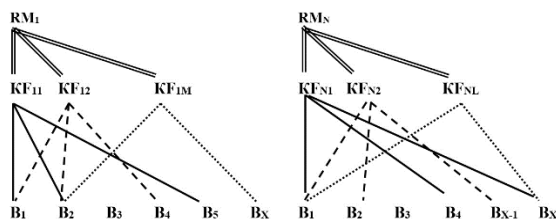


Рис. 8. Общая схема рабочих мест (RM), конфигураций (KF) и виджетов (B)

Каждая конфигурация имеет набор контейнеров, в которые входят преднастроенные виджеты. Реализация отличается от конкретной модели наличием контента, специализированных функций и настроек. Все слои объединены в платформу VSM Cenose [21].

Заключение

Анализ практики использования ситуационных центров и теории синтеза моделей организаций показал сдерживающие факторы развития ситуационных центров: высокая стоимость, длительное время разработки, высокая сложность создания и сопровождения.



Разработанная взаимосвязанная технология автоматизированного синтеза базы данных [17; 18], среднего слоя [19], интерфейса [21] и синтеза СЦ [22] в целом позволяет в 3–5 раз снизить стоимость, время разработки СЦ и решений на его основе.

Список литературы

1. Ильин Н. И., Демидов Н. Н., Новикова Е. В. Ситуационные центры. Опыт, состояние, тенденции развития. М., 2011.
2. Размещение информации о размещении заказов на поставки товаров, выполнение работ, оказание услуг : [официальный сайт]. URL: <http://zakupki.gov.ru/epz/main/public/home.html> (дата обращения: 03.02.2016).
3. Колесников А. В., Меркулов А. А., Майтаков Ф. Г. Технология синтеза ситуационных центров на основе гибридного паттерна // Системный анализ и информационные технологии : шестая междунар. конф. Светлогорск, 2015. Т. 2. С. 92–101.
4. Меркулов А. А. Ситуационный центр VSM Cenose. Калининград, 2014.
5. Колесников А. В., Меркулов А. А. Универсальный паттерн организации ситуационных центров. М., 2013. Т. 23. С. 191–215.
6. Колесников А. В., Меркулов А. А. Программный продукт «Ситуационный центр VSM Cenose». М., 2013. Т. 23. С. 221–241.
7. Что такое компонентно-ориентированное программирование. Информатика-21. URL: <http://www.inr.ac.ru/~info21/info/qtocop.htm> (дата обращения: 05.03.2016).
8. Бек К., Фаулер М. Экстремальное программирование. СПб., 2003.
9. Губко М. В., Коргин Н. А. Классификация моделей анализа и синтеза организационных структур // Управление большими системами : сб. тр. М., 2004. Вып. 6.
10. Камаев В. А., Костерин В. В. Технологии программирования. М., 2006.
11. Новиков А. М., Новиков Д. А. Методология : словарь системы основных понятий. М., 2013.
12. Гнатюк В. И. Закон оптимального построения техноценозов. М., 2005.
13. Бир С. Мозг фирмы. М., 1993.
14. Матурана У. Р., Варела Ф. Х. Древо познания: биологические корни человеческого понимания. М., 2001.
15. Луман Н. Социальные системы. Очерк общей теории. М., 2007.
16. Дмитриевский В. А., Майтаков Ф. Г., Меркулов А. А. Модель данных «Категории сущностей и связей». Калининград, 2014.
17. Дмитриевский В. А., Майтаков Ф. Г., Меркулов А. А. База данных VSM Cenose WEB : патент. Свид. №2015620379 от 26.02.2015. М., 2015.
18. Дмитриевский В. А., Майтаков Ф. Г., Меркулов А. А. Система управления базой VSM Cenose WEB : патент. Свид. № 2015612736 от 26.02.2015. М., 2015.
19. Меркулов А. А., Браценко В. А., Дмитриевский В. А., Несеров П. А. Менеджер плагинов «VSM Cenose» : патент. Свид. 2009613509 от 30.06.2009. М., 2009.
20. Халтурин В. С., Рогачей Н. Н. и др. ГОСТ 24.703-85. Типовые проектные решения в АСУ. М., 1985.
21. Меркулов А. А., Голубков А. В., Дмитриевский В. А. и др. Визонариум VSM Cenose WEB : патент. Свид. № 2015612762 от 26.02.2015. М., 2015.
22. Меркулов А. А., Голубков А. В., Дмитриевский В. А. и др. Ситуационный центр VSM Cenose WEB : патент. Свид. № 2015612764 от 26.02.2015. М., 2015.