

УДК 658.011.56

ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИМИ СИСТЕМАМИ

© 2013 г.

А.В. Запорожцев

Нижегородский государственный технический университет им. Р.Е. Алексеева

zaporozhcev10@mail.ru

Поступила в редакцию 20.06.2013

Рассматриваются вопросы проектирования автоматизированных систем управления организационно-техническими системами. Ключевой вопрос в разработке сложных систем автоматизации – обеспечение целостности системы. Предлагается в качестве логической модели информационной системы использовать архитектуру системы, в которой объединены функциональный, информационный, объектный и организационный взгляды на систему управления. Кроме того, предлагается поэтапный подход к разработке автоматизированной системы с обеспечением целостности системы за счет поддержки её архитектуры в актуальном состоянии на всех этапах разработки.

Ключевые слова: системный подход, проектный подход, автоматизированная система управления, архитектура программной системы, бизнес-процесс.

Системный и проектный подходы в проектировании информационных систем

Разработка больших информационных систем (ИС) требует значительных финансовых и временных затрат. Примером таких систем являются автоматизированные системы управления организационно-техническими системами. Обычно разработка таких систем растягивается на годы, а финансирование производится по годам, что неизбежно приводит к позадачному принципу разработки, когда из контекста общей ИС выделяется отдельная задача, разрабатываемая в следующем финансовом году. Это обстоятельство приводит к «лоскутной автоматизации» на предприятии, негативные последствия которой часто обсуждаются в работах по автоматизации управления [1]. В условиях поэтапной разработки больших ИС принципиальное значение приобретает использование специальной технологии проектирования ИС, обеспечивающей целостность ИС на всех этапах разработки. Базовым элементом такой технологии является использование системного и проектного подходов.

Суть системного подхода в проектировании заключается в соблюдении следующих принципов:

- принцип цели – проектируемая система есть средство достижения реально существующих потребностей системы управления, которые выражаются в цели разработки системы;

- принцип целостности – проектируемая система обеспечит достижение целей разработки системы только в том случае, когда в проекте системы будут учтены все существенные связи между отдельными компонентами системы и связи проектируемой системы с внешними системами.

Проектный подход в разработке ИС сводится к реализации системных принципов на следующих этапах работы:

- определение целей разработки ИС;
- создание рабочей группы проекта;
- разработка проекта ИС и плана-графика реализации проекта;
- разработка программных компонентов ИС;
- поддержание проекта ИС в актуальном состоянии.

Центральным местом в проекте разработки ИС является архитектура ИС [2]. Архитектура ИС – это описание программных комплексов информационной системы и их взаимодействия, что представляет собой логическую модель ИС. Наличие такой логической модели ИС позволяет определить состав программных компонентов ИС, спланировать работы по реализации программной системы ИС и управлять процессами разработки программных компонентов, что открывает возможность поддержания проекта ИС в актуальном состоянии и исключает «лоскутную автоматизацию».

Рабочая группа проекта подчиняется руководителю проекта и является центральным звеном в организации работ. Основной задачей

проектной группы является разработка проекта и поддержание его в актуальном состоянии.

Базовые принципы построения архитектуры ИС

1. Целостность

Центральным требованием к архитектуре ИС является обеспечение целостности системы. Целостность ИС имеет следующие аспекты:

- единая база данных (БД) и единая система справочников; поддержание принципа однократного ввода информации в систему; поддержание целостности данных – требование соответствия данных во взаимосвязанной системе таблиц БД при всех операциях с данными (вставка, корректировка и удаление данных);
- взаимосвязанность программного обеспечения ИС – разделение единой системы на основе бизнес-процессов и объединение программных компонентов ИС на основе использования общих данных и, в первую очередь, на тех данных, которые отражают взаимосвязи бизнес-процессов.

2. Соответствие организационной структуре

Организационная структура любого предприятия разделяется на уровни управления предприятием, а также разделяется на каждом уровне по функциональному признаку на отдельные подразделения, отвечающие за определенный участок работы. Чаще всего целесообразно выделить три уровня управления:

- руководство предприятием;
- среднее звено управления;
- линейный (операционный) уровень.

Разделение организационной структуры по функциональному признаку позволяет разделить бизнес-процесс на отдельные составляющие – компоненты, каждому из которых соответствует одна или несколько функций, подлежащих автоматизации. Разделение необходимо производить на основе ролей, которые выполняет персонал подразделений. Роль – это определенные функциональные обязанности, которые выполняет группа персонала подразделения. Таким образом, организационная структура является вторым фактором, определяющим архитектуру ИС.

3. Соответствие базовым бизнес-процессам

Так как модель деятельности предприятия отражает функции и объекты, используемые в деятельности, то автоматизированная система управления должна основываться на функциях и объектах [3]. Кроме того, современный подход в организации управления предприятием

опирается на разделение деятельности на отдельные бизнес-процессы и управление этими процессами (процессный подход в управлении) [4].

4. Соответствие архитектуре «клиент – сервер»

Современная концепция построения программных систем такого уровня, как ИС, основывается на архитектуре «клиент – сервер» [5]. Суть этой архитектуры в разделении системы на три слоя:

- слой документов – приложения рабочего стола пользователей;
- слой правил бизнеса – приложения обработки данных, общие для различных приложений рабочего стола, которые выполняются на сервере приложений;
- слой управления БД.

Слой документов образует клиентскую часть ИС, а слой правил бизнеса и слой управления БД образуют серверную часть ИС. Такое разделение позволяет сократить затраты на разработку за счет повторно используемых компонентов, унифицировать обработку данных, что исключит появление ошибок при реализации однородных алгоритмов, выполняемых в разных программных комплексах.

При проектировании ИС будут использованы первые два слоя, т.к. слой управления БД относится к реализации ИС.

5. Объектный подход

Практическая деятельность человека всегда связана с совокупностью материальных объектов. Для организационно-технических систем промышленного типа выделяются следующие группы материальных объектов, описание которых может быть необходимым при построении ИС:

1. Структура изделий промышленного производства;
2. Структура технологических цепочек по производству изделий;
3. Структура логистических цепочек (маршруты доставки материальных потоков);
4. Структура производственных помещений, складов и других элементов инфраструктуры предприятия.

Анализ базовых принципов построения архитектуры ИС позволяет сформулировать следующие практические выводы:

1. Базовая структура архитектуры ИС должна основываться на бизнес-процессах. Эта структура определяет логическую, взаимосвязанную совокупность функций, подлежащих автоматизации. Бизнес-процесс состоит из компонентов, каждый из которых представляет совокупность автоматизированных функций – программных модулей (ПМ). ПМ – это «кирпичики» архитектуры ИС.

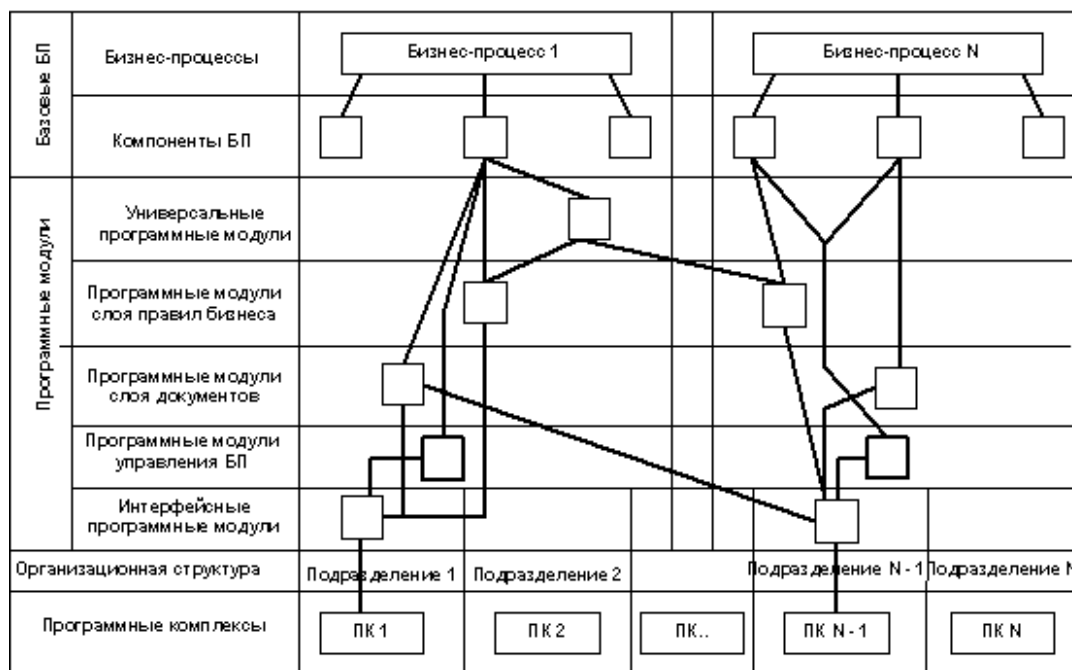


Рис. 1. Концептуальная модель архитектуры ИС

2. ИС делится на программные комплексы на основе организационной структуры. Это позволит определить устойчивый набор программных подсистем ИС, в которых будут объединяться программные модули как автоматизированные функции бизнес-процессов.

3. При разработке программных модулей бизнес-процесса должны быть выделены универсальные программные модули. В таких модулях реализуются те автоматизированные функции, которые нужны в нескольких программных единицах.

Концептуальная модель архитектуры ИС

Базовые принципы построения архитектуры ИС использованы для разработки концептуальной модели архитектуры ИС. Концептуальная модель архитектуры ИС определяет принципы объединения моделей бизнес-процессов и организационную структуру предприятия, а также принципы формирования совокупности программных модулей и их объединения в программные комплексы.

На основе компонентов бизнес-процессов разрабатываются программные модули следующих типов:

1. Универсальные программные модули – модули обработки данных, которые целесообразно сделать универсальными компонентами и использовать в разных программных комплексах ИС.

2. Программные модули слоя правил бизнеса – модули, которые автоматизируют те функции данного бизнес-процесса, которые должны располагаться на сервере ИС.

3. Программные модули слоя документов – модули, которые автоматизируют те функции данного бизнес-процесса, которые относятся к клиентской части ИС.

4. Программные модули управления бизнес-процессами – модули управления бизнес-процессами.

5. Интерфейсные программные модули – модули организации интерфейса пользователя с теми программными модулями, с которыми он должен работать в соответствии со своими должностными обязанностями.

Первые четыре типа программных модулей отражают функциональный аспект деятельности, а интерфейсные программные модули отражают организационный аспект деятельности – распределение функций между персоналом предприятия. Программные модули через интерфейсные программные модули соединяются в программных комплексах информационной системы. Каждый программный комплекс соответствует отдельному функциональному элементу организационной структуры и объединяет совокупность автоматизированных рабочих мест (АРМ) – программных компонентов нижнего, среднего и высшего звена управления. Каждое АРМ ориентировано на конкретную группу персонала, выполняющую однородные обязанности.

Общее проектирование ИС

Анализ предметной области проводится с целью получения необходимой информации о процессах и объектах, относящихся к сфере

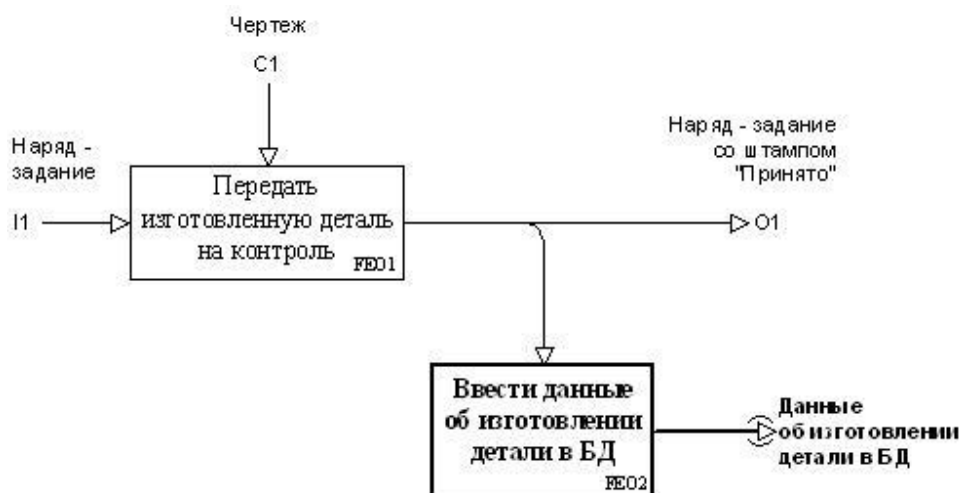


Рис. 2. Выделение автоматизированных функций

действия информационной системы. Сферу действия информационной системы можно представить в виде области, границы которой отделяют процессы и объекты, относящиеся к информационной системе, от тех процессов и объектов предметной области, которые в данной информационной системе не будут представлены. Определение основных контуров сферы действия информационной системы определяется в задании на разработку ИС. На следующих этапах разработки ИС эти границы уточняются исходя из возможностей и целесообразности отображения конкретных процессов и объектов в ИС.

Результаты анализа предметной области должны быть представлены в виде моделей двух типов:

1. Обобщенной модели бизнес-процессов;
2. Семантической информационной модели.

Обобщенная модель бизнес-процессов (ОМБП) должна ответить на основные вопросы относительно будущей информационной системы:

1. С какими внешними системами связана информационная система?
2. Какие базовые бизнес-процессы составят основу информационной системы?
3. Как эти бизнес-процессы между собой связаны?
4. Какие объекты используются этими базовыми бизнес-процессами?

Для описания бизнес-процессов наибольший эффект даст использование методологии системного анализа и проектирования (SADT) [3]. Это методология, в которой предусмотрено разделение входных данных бизнес-процесса на управляющие данные, входные данные и механизмы.

Семантическая информационная модель дает описание характеристик объектов предметной области и их взаимосвязи. Такая модель

послужит основой для проектирования структуры таблиц базы данных информационной системы. Наиболее эффективным инструментом разработки семантической модели предметной области является использование методологии IDEF1X [6]. Семантическая модель, так же как обобщенная модель бизнес-процессов, на всех этапах разработки информационной системы должна поддерживаться в актуальном состоянии.

Архитектура информационной системы объединяет все модели системы в логическую модель и определяет основные программные элементы будущей информационной системы и их взаимодействие. Таким образом, на этапе общего проектирования создается обобщенная модель системы – архитектура системы, которая будет уточняться на следующих этапах проекта. Это обеспечит поддержание целостности модели системы на всех этапах разработки, что является ключевым фактором успеха проекта разработки информационной системы.

Разработка архитектуры системы

Первым шагом построения архитектуры системы является определение автоматизируемых функций. Для этого для каждой функции ОМБП необходимо определить целесообразность ее автоматизации. Для каждой автоматизируемой функции необходимо определить степень автоматизации – будет ли функция автоматизирована полностью или функция будет разделена на автоматизируемую часть и часть, которая будет выполняться без средств автоматизации. В последнем случае необходимо преобразовать ОМБП для выделения функций, которые должны быть полностью автоматизированы. Например, для функции «Изготовить деталь» может быть предложена следующая диаграмма декомпозиции (рис. 2).

Таблица 1

Определение программных модулей

Функции	Входные данные	Выходные данные	Управление	Операции
[A21] Выбрать поставщика	Заявка на материалы	Поставщик	Смета затрат на материалы, поставщики	1. Вести данные по поставщикам 2. Вести данные по материалам 3. Вести данные по параметрам поставщика 4. Выбрать список поставщиков по заданным параметрам
[A22] Определить условия поставки	Заявка на материалы	План-график поставки материалов	Смета затрат на материалы, поставщик	1. Формировать сводную ведомость поставки материалов 2. Формировать план-график поставки материалов 3. Корректировать план-график поставки материалов
[A23] Обеспечить доставку материалов		Путевой лист	План-график поставки материалов	1. Формировать путевой лист
[A24] Контролировать поставку материалов (объемы, качество, сроки)	Накладная на материал	Заверенная накладная на материалы, отчет о поставке материалов	Путевой лист	1. Регистрировать накладную на поставку материалов 2. Формировать отчет о поставке материалов
[A25] Подготовить данные по снабжению	Отчет о поставке материалов, табель рабочего времени	Данные о снабжении	Показатели, заказ поставки материалов, путевой лист, накладные расходы	1. Формировать сводный отчет о работе отдела снабжения

В данном примере введена дополнительная функция «Вести данные об изготовлении детали в БД», которая будет автоматизирована, а наряд-задание останется бумажным документом, используемым в процедуре изготовления детали.

Для каждой автоматизируемой функции необходимо определить состав операций, реализующих эту функцию. Это можно сделать путем декомпозиции функций в функциональной модели (что является лучшим вариантом) или путем определения операций в специальной таблице. В качестве примера рассмотрим определение операций бизнес-процесса «Управлять снабжением» (табл. 1).

Данные для столбцов «Функция», «Входные данные», «Выходные данные», «Управление» берутся из функциональной модели бизнес-процесса «Управлять снабжением», а столбец «Операции» заполняется исходя из требований обработки данных.

Следующим шагом является формирование программных комплексов на основе выделенных операций. Эта процедура производится на основе организационной структуры, в рамках которой функционирует ИС. Организационная структура может быть разделена как по уровням управления, так и по функциям внутри подразделения. Минимальная совокупность

функций персонала, которая рассматривается в архитектуре информационной системы, называется ролью. Например, отдел снабжения может быть представлен следующими ролями:

1. Инженер по работе с поставщиками и материалами;
2. Инженер-плановик;
3. Инженер по доставке материалов.

На основе списка операций и списка ролей необходимо составить матрицу ответственности (табл. 2).

Анализ этой матрицы позволяет проконтролировать распределение операций и принять решение по определению АРМ и программных модулей. На этапе разработки архитектуры ИС целесообразно выделять логически обоснованные АРМ, а на этапе реализации проекта это разделение может быть уточнено или в сторону разделения АРМ на части, или за счет объединения АРМ в более крупные программные комплексы. Для рассматриваемого примера были выделены следующие АРМ: «Поставка», «Плановик», «Доставка».

При формировании программных модулей необходимо ориентироваться на однородность автоматизируемых функций. Под однородностью понимается близость по назначению функции и способу реализации. Однородные

Таблица 2

Распределение программных модулей по ролям

№	Операции	Инженер по работе с поставщиками	Инженер-плановик	Инженер по доставке материалов
1	Вести данные по поставщикам	X		
2	Вести данные по материалам	X		
3	Вести данные по параметрам поставщика	X		
4	Выбрать список поставщиков по заданным параметрам	X		
5	Формировать сводную ведомость поставки материалов		X	
6	Формировать план-график поставки материалов		X	
7	Корректировать план-график поставки материалов		X	
8	Формировать путевой лист			X
9	Регистрировать накладную на поставку материалов			X
10	Формировать отчет о поставке материалов			X
11	Формировать сводный отчет о работе отдела снабжения		X	

Таблица 3

Определение программных модулей

№	Операции	АРМ	ПМ
1	Вести данные по поставщикам	Поставка	Учет поставщиков и материалов
2	Вести данные по материалам		
3	Вести данные по параметрам поставщика		Отчет по поставщикам
4	Выбрать список поставщиков по заданным параметрам	План	Сводная ведомость поставок
5	Формировать сводную ведомость поставки материалов		
6	Формировать план-график поставки материалов		План-график поставок
7	Корректировать план-график поставки материалов	Доставка	Документы по доставке
8	Формировать путевой лист		
9	Регистрировать накладную на поставку материалов		Отчет по доставке
10	Формировать отчет о поставке материалов	Поставка	Сводный отчет о работе отдела
11	Формировать сводный отчет о работе отдела снабжения		

функции целесообразно объединить в одном программном модуле, что позволит их реализовать с меньшими затратами. В рассматриваемом примере функции: «Вести данные по поставщикам», «Вести данные по материалам», «Вести данные по параметрам поставщика» были объединены в один программный модуль «Учет поставщиков и материалов», а функция «Формировать сводную ведомость поставки материалов» образовала отдельный программный модуль «Отчет по поставщикам». Для рассматриваемого примера были сформированы программные модули, которые представлены в таблице 3.

Разделение программных модулей по типам (универсальные программные модули, программные модули слоя правил бизнеса, программные модули слоя документов, программные модули управления бизнес-процессом и интерфейсные программные модули) производится на этапе разработки программной системы.

Завершает первый этап проектирования разработка плана-графика реализации проекта. План-график определяет последовательность разработки отдельных подсистем ИС – программных компонентов одного бизнес-процесса. При составлении плана-графика учитывается как приоритет в разработке отдельных компонентов программной системы, так и наличие ресурсов для разработки. Поэтапный характер разработки больших информационных систем потребует систематического пересмотра плана-графика. Прежде всего это связано с тем, что на каждом этапе разработки будут уточняться как общие, так и отдельные элементы проекта – список автоматизированных функций, дополнительные элементы информационной модели, новые требования и т.д. Кроме того, реальная жизнь (задержки в разработке, ограниченность ресурсов и т.д.) внесет свои коррективы в планы разработки. Все эти факторы должны учиты-

ваться не только в базовых элементах проекта (модели бизнес-процессов, информационной модели и архитектуре системы), но и в организационном документе проекта – плане-графике.

Проектирование отдельной подсистемы ИС

Проектирование отдельной подсистемы представляет собой процесс определения детального состава программных модулей подсистемы и уточнение таблиц баз данных, с которыми будут работать эти программные модули. Основной задачей проектирования отдельной подсистемы ИС является детальная проработка модели бизнес-процессов и информационной модели для отдельной подсистемы ИС.

Необходимость детальной проработки бизнес-процессов связана с тем, что обобщенная модель бизнес-процессов, по которой разрабатывалась архитектура ИС, является самым общим представлением, необходимым только для определения основных подсистем ИС. При проектировании конкретной подсистемы ИС необходимо более детальное описание бизнес-процессов, раскрывающее содержание основных функций подсистемы и уточняющее состав этих функций.

В ходе разработки моделей бизнес-процессов обычно возникает предложение по модификации обобщенной модели бизнес-процессов. Такие предложения могут касаться изменения состава операций бизнес-процессов и используемых объектов. Предложения по корректировке обобщенной модели бизнес-процессов и по включению новых объектов направляются в рабочую группу, занимающуюся общим проектированием ИС. Необходимость этой работы обусловлена тем, что только в обобщенной модели можно учесть взаимосвязи между разными бизнес-процессами, а поддержка такой обобщенной модели в актуальном состоянии гарантирует целостность ИС.

Проработка информационной модели заключается в уточнении сущностей и атрибутов, которые необходимо учесть для согласования информационной модели с откорректированной моделью бизнес-процессов. На этапе проектирования отдельной подсистемы решаются вопросы реализации объектной модели предметной области и разделения модулей на модули слоя документов и модули слоя правил бизнеса.

Кроме этого, необходимо перейти от семантической модели предметной области к физической информационной модели, в которой отображены решения о способе хранения информации в таблицах базы данных информационной системы. В дальнейшем в актуальном состоянии необходимо поддерживать и семантическую, и физическую информационную модель и их соответствие друг другу.

Результатом детальной проработки модели бизнес-процессов и информационной модели для отдельной подсистемы ИС является уточненный список программных модулей и их операций, а также описание таблиц базы данных системы.

Заключение

Обеспечение целостности больших информационных систем требует поддержания моделей системы в актуальном состоянии на всех этапах разработки. Предлагается разделить проектирование информационных систем на два этапа: этап общего проектирования и этап проектирования отдельных подсистем. На этапе общего проектирования формируется архитектура системы как логическая модель информационной системы, объединяющая модель бизнес-процессов, информационную модель и модель организационной структуры. На этапе детального проектирования отдельных подсистем происходит уточнение архитектуры системы, что обеспечивает поддержание моделей системы в актуальном состоянии.

Список литературы

1. Баронов В.В., Калянов Г.Н., Попов Ю. Н., Титовский И.Н. Информационные технологии и управление предприятием. М.: Компания АйТи, 2009. 328 с.
2. Макгрегор Дж. Программная архитектура // Открытые системы. 2004. № 08.
3. Марка Д.А., Мак-Гоуэн К. Методология структурного анализа и проектирования. М.: Весть-МетаТехнология, 1999. 240 с.
4. Калянов Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов. М.: Финансы и статистика, 2006. 240 с.
5. Васкевич Д. Стратегия клиент/сервер. Руководство по выживанию для специалистов по реорганизации бизнеса. К.: Диалектика, 1996. 384 с.
6. Маклаков С.В. ВРwin и ERwin. CASE – средства разработки информационных систем. М.: ДИАЛОГ-МИФИ, 2000. 256 с.

**DESIGN PROBLEMS OF AUTOMATIC SYSTEMS
TO CONTROL ORGANIZATIONAL-TECHNICAL SYSTEMS**

A.V. Zaporozhcev

The article considers designing automatic control systems for organizational-technical systems. The key problem in the development of complex automatic systems is to ensure system integrity. A logical model of the automatic control system is proposed whose architecture combines functional, informational, object, and organizational approaches to the system management. Step-by-step procedures are also proposed to ensure system integrity by maintaining the system architecture in its actual state at all stages of the system development.

Keywords: system approach, project approach, automated control system, software architecture, business process.