

### Выводы

1. Предложена адаптивная система регулирования температуры углеводородного конденсата на выходе из кожухотрубного теплообменника, включающая ПИ-регулятор и последовательное псевдолинейное корректирующее устройство динамических свойств систем автоматического регулирования.

### СПИСОК ЛИТЕРАТУРЫ

1. Солдатов В.В., Ухаров П.Е. Адаптивная настройка систем управления с ПИД-регуляторами в условиях информационной неопределенности // Приборы и системы. Управление, контроль, диагностика. – 2004. – № 8. – С. 16–20.
2. Штейнберг Ш.Е., Залуцкий И.Е., Сережин Л.П., Варламов И.Г. Настройка и адаптация автоматических регуляторов. Инструментальный комплект программ // Промышленные АСУ и контроллеры. – 2003. – № 10. – С. 43–47.
3. Скороспешкин М.В. Адаптивные псевдолинейные корректоры динамических характеристик систем автоматического регу-

2. Экспериментально показана эффективность предложенной системы регулирования при изменении во времени параметров объекта управления.
3. Применение предложенного корректирующего устройства позволило реализовать систему регулирования объектами с нестационарными параметрами. Устройство можно добавлять в действующие системы регулирования на базе микропроцессоров без дополнительных затрат на аппаратную часть.

лирования // Известия Томского политехнического университета. – 2006. – Т. 309. – № 7. – С. 172–176.

4. Скороспешкин М.В., Цапко Г.П. Адаптивный корректор динамических характеристик систем автоматического регулирования // Радиоэлектроника, электротехника и энергетика: Труды XII Междунар. научно-технич. конф. студентов и аспирантов. – Т. 1. – М.: МЭИ, 2006. – С. 498–499.

Поступила 01.04.2010 г.

УДК 669.162.28

## ТЕХНОЛОГИЯ И СРЕДСТВА РАЗРАБОТКИ ИНФОРМАЦИОННО-МОДЕЛИРУЮЩИХ СИСТЕМ ДЛЯ РЕШЕНИЯ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ В МЕТАЛЛУРГИИ

Н.А. Спирин, В.В. Лавров, А.А. Бурькин, А.В. Краснобаев\*, А.Г. Быков

ГОУ ВПО «Уральский государственный технический университет – УПИ  
имени первого Президента России Б.Н. Ельцина», г. Екатеринбург  
\*ОАО «Магнитогорский металлургический комбинат», г. Магнитогорск  
E-mail: lv@tim.ustu.ru

*Отражены технологические особенности и средства разработки программного обеспечения, использованные авторами в ходе создания современных информационно-моделирующих систем для решения технологических задач в области доменного производства, в частности решения задачи оптимального распределения топливно-энергетических ресурсов в группе доменных печей.*

### Ключевые слова:

*Технология разработки программного обеспечения, система поддержки принятия решений, доменная печь, технологические задачи в металлургии.*

### Key words:

*Software engineering, decision support systems, blast furnace, technological problems in metallurgy.*

В настоящее время все более очевидной становится роль алгоритмов и компьютерных программ для решения комплекса технологических задач в области металлургии MES-уровня (*Manufacturing Execution Systems* – системы управления технологией, производственными процессами) современных автоматизированных информационных систем крупнейших металлургических предприятий России. Это определяет потребность в разработке специализированного программного обеспечения информационно-моделирующих систем, в основу которого положен комплекс математических моделей, учитывающих как физику процесса, основы

теории тепло- и массообмена, законы сохранения энергии, так и особенности влияния технологических и стандартных характеристик сырья на показатели производственного процесса. При этом важно обеспечить высокий уровень их интеграции с существующими производственными и корпоративными системами.

Значительную роль в успешном внедрении и использовании информационно-моделирующих систем играет качество разработанного программного обеспечения. Среди наиболее значимых показателей качества современных программных средств выделены функциональность, надежность,

легкость применения и сопровождаемость. Указанные показатели фиксируются во внешнем описании программного обеспечения, которое разрабатывается на основе требований заказчиков. Разработка качественного программного обеспечения информационно-моделирующих систем, невозможна без использования современных технологических подходов и компьютерных инструментальных средств.

Авторами накоплен практический опыт в ходе разработки программного обеспечения компьютерных модельных систем аглодоменного производства ОАО «Магнитогорский металлургический комбинат», позволяющий более качественно использовать существующие на комбинате информационные ресурсы для анализа и прогнозирования производственных ситуаций [1–3]. В основу технологического подхода к разработке программного обеспечения положена итерационная (спиральная) модель, приводящая к выпуску внутренней или внешней версии программного изделия, которое в дальнейшем совершенствуется от итерации к итерации, чтобы стать законченной системой (рис. 1).

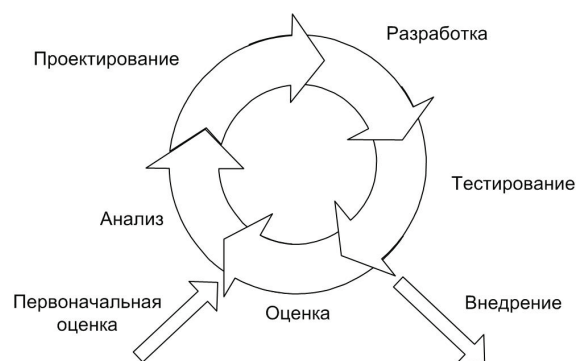


Рис. 1. Итерационный технологический подход к разработке программных продуктов

На каждой итерации реализуется часть функционала конечной системы. Каждая итерация представляет из себя законченный цикл разработки продукта, с присущими ему стадиями: анализ требований, проектирование, реализация, отладка, тестирование и опытно-промышленная эксплуатация. Результатом каждой итерации является целостный продукт, который передается пользователям для опытно-промышленной эксплуатации. Основными преимуществами итерационного процесса перед каскадным являются:

- существенно упрощается внесение изменений в проект при изменении требований заказчика;
- уменьшение уровня рисков;
- возможность внесения тактических изменений в проект;
- возможность использования перспективных технологических подходов к программированию.

Каждый из перечисленных процессов разработки характеризуется определенными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами.

**Проектирование.** При проектировании систем используется объектно-ориентированный подход.

Объектно-ориентированный подход к проектированию представляет собой современную методологию проектирования, соединяющую в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы [4–7]. Данный подход подразумевает представление системы в виде группы взаимодействующих объектов, каждый из которых представляет некую сущность моделируемой предметной области и характеризуется классом, состоянием и поведением. В процессе проектирования для описания системы с различных точек зрения используются три типа моделей: классов, состояний и взаимодействия.

Модель классов описывает статическую структуру объектов системы и их отношения, определяет контекст разработки программы, то есть предметную область. Цель конструирования классов состоит в том, чтобы охватить те реальные концепции, которые существенны для программного приложения. Модель классов изображается на диаграммах классов.

Модель состояний описывает аспекты объектов, связанные с течением времени и с последовательностью операций, то есть события, связанные с изменениями, состояния, определяющие контекст событий, и упорядочение событий и состояний. Модель состояний описывает последовательности операций, происходящих в системе в ответ на внешние воздействия. Модель состояний охватывает вопросы управления – аспект системы, описывающий порядок осуществляемых операций без учета их фактического значения, участников и реализации. Эта модель реализуется посредством диаграмм состояний.

Модель взаимодействия описывает кооперацию объектов системы для обеспечения необходимого поведения системы как целого. Построение модели начинается с анализа вариантов использования приложения, которые затем уточняются на диаграммах последовательности и диаграммах деятельности. Вариант использования описывает функциональность системы, то есть то, что система делает для пользователей. Диаграмма последовательности изображает взаимодействие объектов и временную последовательность этого взаимодействия.

Три описанные модели являются связанными между собой составляющими полного описания системы. Для создания и документирования моделей используется нотация UML [8].

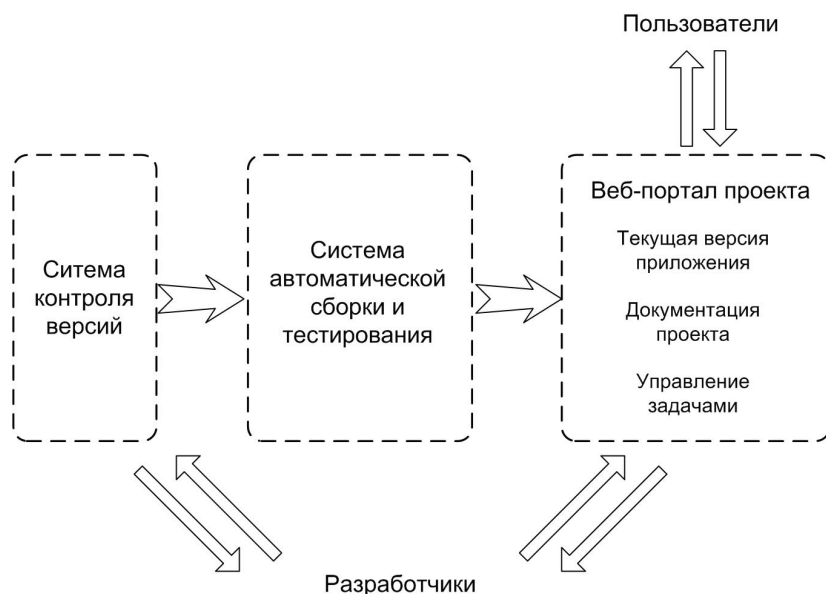


Рис. 2. Процесс разработки программного продукта

**Разработка.** В процессе реализации проектных решений используются системы контроля версий, управления задачами и портал проекта (рис. 2).

Система контроля версий позволяет организовать совместную работу группы разработчиков над одним и тем же проектом. Система контроля версий содержит последнюю версию исходных кодов проекта и позволяет одновременно вносить изменения в исходные коды проекта разными разработчиками. В качестве системы контроля версий используется среда Subversion, к основным преимуществам которой относятся:

- возможность отслеживания версии не только файлов, но и каталогов;
- публикация изменений в нескольких файлах и каталогах, как единой транзакции. Это значит, что либо в хранилище попадают все изменения, либо состояние хранилища не изменяется;
- передача между клиентом и сервером только различий в файлах при любых обновлениях версий;
- поддержка копирования, перемещения и переименования файлов с сохранением истории изменений;
- возможность задания любому файлу и каталогу произвольный набор свойств, состоящих из названия и значения. Свойства тоже находятся под управлением версиями;
- возможность одинаково эффективной работы как с текстовыми, так и с двоичными файлами;
- свободное распространение системы, лицензия аналогична Apache/BSD.

Портал проекта содержит систему управления задачами и систему ведения документации.

Система документации содержит утвержденную ранее проектную документацию. Система ведения документации основана на системе Вики (*Wiki*). Вики – гипертекстовая среда (обычно веб-сайт) для сбора и структурирования письменных сведений пользователей. Характеризуется следующими признаками:

- множество авторов. Система управления доступом к материалам;
- возможность многократно править текст посредством самой вики-среды (веб-сайта), без применения особых инструментов на стороне редактора;
- проявление изменений сразу после их внесения;
- разделение информации на отдельные страницы, где у каждой есть своё название;
- особый язык разметки, позволяющий легко и быстро размечать в тексте структурные элементы, форматирование, гиперссылки, списки и т. п.
- учёт изменений (учёт версий) текста и возможность отката к ранней версии.

Система управления задачами позволяет планировать процесс разработки программного продукта, учитывать и контролировать ошибки и следить за процессом устранения этих ошибок. Первоначально в систему управления задачами заносятся задачи, которые нужно решить для реализации программного продукта. По мере разработки продукта в систему управления задачами помещается информация об обнаруженных ошибках. Так же в эту систему помещаются «заявки» от пользователей – как сообщения об ошибках и неудобствах, так и запросы на добавление нового функционала.

Главный компонент системы управления задачами — база данных, содержащая сведения о задачах:

- автор задачи;
- дата и время, когда была добавлена задача;
- важность задачи;
- описание задачи;
- кто занимается решением задачи;
- состояние задачи;
- прикрепленный файл, например файл с изображением.

В процессе разработки используется система Trac, которая совмещает систему ведения документации и систему управления задачами. К основным достоинствам системы Trac относится мощная система управления ошибками, наличие движка вики, тесная интеграция с системой контроля версий Subversion, расширяемая архитектура, наличие множества готовых модулей расширений, лицензия — модифицированная BSD лицензия.

На этапе реализации проекта авторами используется принцип непрерывной интеграции. Непрерывная интеграция (англ. — *Continuous Integration*) — термин, относящийся к разработке программного обеспечения и обозначающий автоматизированный процесс, выполняющий частые пересборки и тесты приложения. Практически это выглядит как отдельный процесс, запущенный на сервере, который следит за изменениями на файловой системе либо в системе управления версиями и автоматически запускает полную пересборку всех модулей приложения и прогон тестов.

К основным преимуществам непрерывной интеграции относятся:

- выявление и исправление проблем интеграции непрерывно, а не в самом конце разработки;
- ранние предупреждения об испорченном/несовместимом коде;
- немедленное юнит-тестирование всех изменений;
- постоянное наличие «текущей» собранной версии — для тестирования, демонстрации, других применений.

В большинстве проектов использована система CruiseControl.Net, которая представляет из себя автоматизированный сервер непрерывной интеграции. К основным преимуществам этой системы относятся:

- работа с различными системами контроля версий;
- работа с различными системами сборки проектов;
- работа с различными системами тестирования;
- наличие web-приложения для отслеживания статуса и детального отчета о сборке проектов;
- свободная лицензия, схожая с лицензиями Apache и BSD.

**Отладка и тестирование.** Для автоматизированного тестирования продукта после сборки, как правило, разрабатывается набор юнит-тестов. Юнит-тестирование (англ. — *unit test*) — это процесс, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Цель юнит-тестирования — изолировать отдельные части программы и показать работоспособность отдельных частей приложения. В качестве среды юнит-тестирования используется система NUnit.

При разработке информационно-моделирующих систем применялась трехзвенная архитектура (рис. 3). В отличие от «классической» двухзвенной архитектуры «клиент-сервер», в трехзвенной архитектуре, помимо клиента и сервера баз данных, присутствует сервер приложений, выполняющий роль промежуточного звена.

Трехзвенная архитектура включает в себя уровни программного обеспечения, реализующих следующие функции системы (рис. 3):

1. Представление. Предоставление графического интерфейса пользователя и для обеспечения доступа к функциональности системы. Данный уровень может быть реализован как на основе web-страниц, так и на основе «облегченных» версий традиционных приложений.
2. Бизнес-логика. На этом уровне реализуется вся функциональность системы. Он предназначен для извлечения и преобразования данных. Данный уровень реализуется с помощью таких технологий, как RPC, CORBA, DCOM, Java EE и др. Однако, в последнее время все большее распространение получают SOAP и веб-сервисы.
3. Данные. Уровень предназначен для хранения данных. Данный уровень реализуется с помощью систем СУБД.

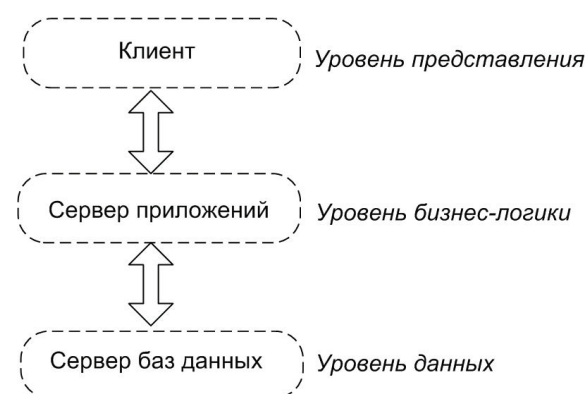


Рис. 3. Трехзвенная архитектура информационно-моделирующей системы

Такая архитектура имеет ряд преимуществ:

- масштабируемость;
- конфигурируемость — изолированность уровней друг от друга — позволяет быстро и простыми средствами переконфигурировать систему



при возникновении сбоев или при плановом обслуживании на одном из уровней;

- повторное использование программных модулей – модуль размещенный на сервере приложений может использоваться одновременно многими пользователями в составе различных приложений;
- простота интеграции в существующие корпоративные информационные системы;
- высокая безопасность;
- высокая надежность;
- низкие требования к производительности и техническим характеристикам клиентов, как следствие снижение их стоимости.

С учетом описанной выше технологии и средств разработки разработана архитектура информационно-моделирующей системы для оптимального распределения топливно-энергетических ресурсов в группе доменных печей, архитектура построения которой отражена на рис. 4. Оптимальное распределение топливно-энергетических ресурсов, в частности инжектируемого топлива и кислорода, в пределах группы доменных печей является актуальной задачей, поскольку технологические показатели работы отдельных печей существенно различаются. Наличие многих факторов и критериев, определяющих эффективность использования комбинированного дутья, а также ограничений на расходы топливно-энергетических ресурсов, существенно усложняет задачу по определению оптимальных параметров дутья, при которых достигаются наилучшие технико-экономические показатели работы, как отдельных доменных печей, так и группы печей или цеха в целом.

При заданных на доменный цех общих расходах инжектируемого топлива и кислорода целесообразно иметь оперативную методику оценки эффективности использования указанных ресурсов на доменных печах и осуществлять их оптимальное распределение. Постановка задачи и основные подходы к ее решению достаточно полно опубликованы в [1, 2] и поэтому здесь не приводятся.

Основу системы составляет сервер приложений. Он выполняет расчеты по оптимизации распределения ТЭР, основываясь на запросах клиента и данных из БД. Сервер приложений включает в себя модули:

- расчет коэффициентов целевой функции;
- формирование ограничений на цех;
- формирование технологических ограничений на печи;
- модуль оптимизации.

Сервер приложений создан на базе Microsoft IIS 7 и технологии .NET Framework. Доступ к функциональности сервера приложений осуществляется с помощью веб-сервисов по протоколу SOAP.

База данных расположена на сервере баз данных Microsoft SQL Server 2005 и содержит усредненные показатели работы доменного цеха, получаемые из существующих систем сбора данных (КИС, Сервера АСУТП и др.).

Клиентское приложение выполнено в виде традиционного Windows приложения на основе технологии .NetFramework. Основными его функциями являются:

- проверка вводимых данных;
- визуализация данных;
- формирование отчетов;
- вызов расчетных процедур на сервере приложений.

Разработанное на основе оптимизационной экономико-математической модели программное обеспечение позволяет прогнозировать значения расходов природного газа и кислорода на каждую из печей с целью получения максимальной выгоды от использования комбинированного дутья при изменении объема выделенных цеху ресурсов, при остановке одной или нескольких печей, при изменении режимных и сырьевых параметров работы отдельных печей и при других технологических ситуациях. Для каждой доменной печи при оптимальной подаче природного газа и кислорода возможно определение следующих показателей ее работы:

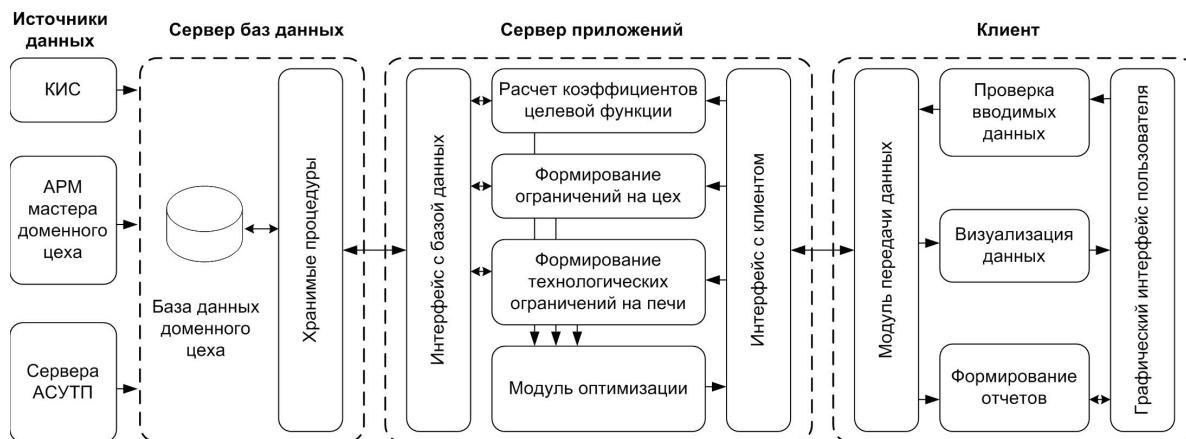


Рис. 4. Архитектура информационно-моделирующей системы оптимального распределения ТЭР в группе доменных печей

- расход природного газа, м<sup>3</sup>/ч;
- расход кислорода, м<sup>3</sup>/ч;
- расход кокса, т/ч;
- производительность печи, т/ч;
- температура горения на фурмах, °С;
- содержание кремния в чугуна, %;
- содержание серы в чугуна, %;
- удельные затраты тепла в нижней зоне печи, МДж/т чугуна;
- отношение теплоемкостей потоков в шахте, доли;
- степень уравнивания шихты, доли;
- эффективности использования газа и кислорода, р/ч.

Программное обеспечение предусматривает решение задачи оптимального распределения природного газа и кислорода для двух периодов работы доменных печей: базового и проектного.

Расчет для базового периода производится по фактическим исходным данным, отражающим уже прошедший период работы доменных печей. В этом случае пользователь с помощью программы может оценить, насколько эффективно был использован природный газ и кислород в этом уже состоявшемся периоде. В частности, можно рассчитать для прошедшего периода оптимальный расход природного газа и кислорода на каждую печь, определить показатели работы печей при этих расходах и выполнить сравнительный анализ всех вышеперечисленных показателей при произво-

дешней (базовой) и оптимальной подаче природного газа и кислорода.

Расчет по программе для проектного периода можно использовать для определения оптимального распределения природного газа, кислорода и показателей работы в проектный период, когда предполагается изменение дутьевых параметров работы отдельных печей. В этом случае будут рассчитаны оптимальные значения показателей работы для каждой доменной печи.

#### Выводы

Показано, что применение современных технологий, средств и методик разработки программных продуктов позволяет создавать функциональные, надежные, легкие в применении, сопровождаемые, интегрируемые системы с минимальными рисками и в приемлемые сроки. Использование разработанной информационно-моделирующей системы в АСУ доменной плавки на ОАО «Магнитогорский металлургический комбинат» позволяет решать оперативные задачи управления технологией доменной плавки, обеспечивает повышение эффективности принятия решений инженерно-техническим персоналом в условиях изменений объема топливно-энергетических ресурсов, нестабильности состава и качества проплавляемого железорудного сырья и конъюнктуры рынка.

*Работа выполнена в соответствии с Государственным контрактом Федерального агентства по науке и инновациям № 02.740.11.0152.*

#### СПИСОК ЛИТЕРАТУРЫ

1. Онорин О.П., Спиринов Н.А., Терентьев В.Л. и др. Компьютерные методы моделирования доменного процесса / под ред. Н.А. Спирина. – Екатеринбург, УГТУ-УПИ, 2005. – 301 с.
2. Спиринов Н.А., Лавров В.В., Паршаков С.И., Денисенко С.Г. Оптимизация и идентификация технологических процессов в металлургии / под ред. Н.А. Спирина. – Екатеринбург: УГТУ-УПИ, 2006. – 307 с.
3. Спиринов Н.А., Ипатов Ю.В., Лобанов В.И. и др. Информационные системы в металлургии / под ред. Н.А. Спирина. – Екатеринбург: УГТУ-УПИ, 2001. – 617 с.
4. Брауде Э.Дж. Технология разработки программного обеспечения [пер. с англ.]. – СПб.: Питер, 2004. – 655 с.
5. Гамма Э., Хелм Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования [пер. с англ.]. – СПб.: Питер, 2007. – 368 с.
6. Фаулер М. Архитектура корпоративных программных приложений [пер. с англ.]. – М.: Вильямс, 2006. – 544 с.
7. Макконнелл С. Совершенный код. Мастер-класс [пер. с англ.]. – М.: Русская редакция; СПб.: Питер, 2007. – 896 с.
8. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. – СПб.: Питер, 2006. – 544 с.

*Поступила 15.01.2010 г.*