

А.А. ЗЕМЛЯНОЙ, аспирант, НТУ "ХПИ" (г. Харьков),
Н.В. ТКАЧУК, д-р. техн. наук, НТУ "ХПИ" (г. Харьков),
Р.А. ГАМЗАЕВ, аспирант, НТУ "ХПИ" (г. Харьков)

КОМПЛЕКС МОДЕЛЕЙ ТЕХНОЛОГИИ ПРОТОТИПИРОВАНИЯ ДЛЯ АДАПТИВНОГО ПРОЕКТИРОВАНИЯ КОМПОНЕНТНЫХ ПРОГРАММНЫХ РЕШЕНИЙ

У процесі адаптивної розробки програмного забезпечення його архітектура й компоненти перетерплюють зміни в міру зміни пропонованих системних вимог. У статті представлений новий підхід до адаптивного проектування, заснований на технології прототипування. Представлені моделі й технології дозволяють створювати, конфігурувати, виконувати й аналізувати прототипи системи, що розроблюється, з метою зменшення ресурсів, необхідних для її розробки.

В процессе адаптивной разработки программного обеспечения его архитектура и компоненты претерпевают изменения по мере изменения предъявляемых системных требований. В статье представлен новый подход к адаптивному проектированию, основанный на технологии прототипирования. Представленные модели и технологии позволяют создавать, конфигурировать, выполнять и анализировать прототипы разрабатываемой системы с целью уменьшения ресурсов, необходимых для ее разработки.

During the adaptive development process software components and system architecture constantly change with respect to corresponding changes in system requirements. In the paper we present a new design approach for adaptive software development which is based on prototyping technology. Models and technologies integrated into the approach provide processes of system prototypes building, configuring, execution and estimation aiming to reduce development costs and needed resources.

1. Постановка проблемы, цели и актуальность исследования. В настоящее время для проектирования, разработки и сопровождения сложного программного обеспечения (ПО) широко используются гибкие, ориентированные на повторное использование компонентов технологии. Гибкость процессов и повторное использование компонентов особенно важны в случаях, когда программная система разрабатывается с учетом существующих унаследованных систем, в условиях постоянно меняющейся внешней среды системы и предъявляемых к ней системных требований (СТ). Проблемой в таких условиях является то, что разрабатываемая система должна в полной мере соответствовать изменяющимся СТ. Разработка моделей и технологий, обеспечивающих возможность такой адаптации и предполагающей выполнение ряда критериев качества и стоимости по-

лучаемых программных решений, являются весьма актуальной научно-технической проблемой [1].

Учитывая современные тенденции в разработке ПО и основываясь на опыте разработки сложных распределенных информационно-управляющих систем (ИУС) в ходе выполнения проектов автоматизации объектов добычи нефти и газа в Харьковской области [2,3] была разработана макро-технология адаптивного проектирования (МТАП) компонентных программных решений (КПР), используемая в процессах адаптивного проектирования ИУС [3]. Одной из важных составляющих этого подхода является предложенная в [4] интегрированная технология эволюционного прототипирования (ИТЭП) КПР.

Цель данной работы - обосновать методологические подходы к созданию ИТЭП и представить модели, разработанные для формализации ее основных этапов.

2. Обзор современных средств и методов прототипирования ПО. Прототипирование как один из эффективных подходов к моделированию сложных систем используется для согласования требований с заказчиком, определения видения будущей системы, а так же для получения и сбора отзывов о системе. Прототипирование обычно используется, когда СТ полностью не ясны, либо не могут быть сразу определены. Среди наиболее часто используемых методов прототипирования можно выделить: *быстрое прототипирование* (rapid prototyping или throwaway prototyping), *эволюционное прототипирование* (evolutionary prototyping) и инкрементное *прототипирование* (incremental prototyping). При этом следует отметить, что одной из наиболее важных целей каждого из методов прототипирования является задача валидации СТ. На рис. 1 представлена одна из возможных классификаций моделей и инструментальных средств, используемых при этих подходах.

Быстрое прототипирование применяется с целью уменьшения количества рисков связанных со сбором СТ, при этом созданная модель системы практически никогда не входит в окончательную версию программного продукта.

Целью эволюционного прототипирования является создание надежного прототипа системы в структурированной форме и постоянное его совершенствование. При разработке системы с использованием эволюционного прототипирования ПО на каждом шаге совершенствуется и перестраивается. Этот метод позволяет команде разработчиков добавлять новые возможности в систему, а так же вносить изменения, которые могут быть не видны на этапе сбора системных требований. Преимущество эво-

люционного прототипирования по сравнению с быстрым прототипированием в том, что в ходе него появляется функциональная система, она может быть использована как начальный базис для будущей системы.

Инкрементное прототипирование подразумевает, что конечный продукт в целом создается как набор прототипов. Такой подход позволяет выявлять сложности в реализации целевой системы еще на ранних стадиях ее создания.

Для создания прототипов существуют различные системы, одной из первых была Computer-Aided Prototyping System (CAPS), разработанная в Школе повышения квалификации ВМС США (Naval Postgraduate School) [5], для быстрого прототипирования встроенных систем реального времени, таких как системы наведения ракет, космических челночных систем авионики, и т.д. В ней используется эволюционный подход к прототипированию, основанный на языке описания прототипов (PSDL). Он используется для анализа СТ, технико-экономического обоснования и разработки больших встроенных систем.

Система SLAM [6] позволяет эффективно использовать формальные методы для быстрой разработки программного обеспечения. Эта система включает выразительный объектно-ориентированный язык, спецификации и среду разработки, которая среди прочего, может создавать код на языках высокого (Java, C++ и т.д.). Дополнительным преимуществом является то, что используемый метод прототипирования является эволюционным и отдельные его части могут быть напрямую использованы в конечной системе.

Axure RP Pro [7] - это программный продукт позволяющий быстро создавать прототипы пользовательского интерфейса для Web-приложений без написания кода, создавать структурные схемы страниц, диаграммы и спецификацию к Web-сайтам. iRise Studio [8] - средство используемое бизнес-аналитиками и руководителями проектов, для быстрого определения и визуализации программного обеспечения. iRise может быть использован для создания полностью интерактивной модели целевого применения системы, эта модель создается в интерактивном режиме и не требует написания программного кода.

Хочется отметить, что почти все перечисленные системы прототипирования, в основном используются для моделирования и валидации функциональных СТ. Такие инструменты как CAPS основываются на использовании методов формализации и применяются для исследования систем реального времени, а Axure RP и iRise созданы для быстрого прототипирования пользовательского интерфейса, что может служить основой для дальнейшего совершенствования требований к системе.

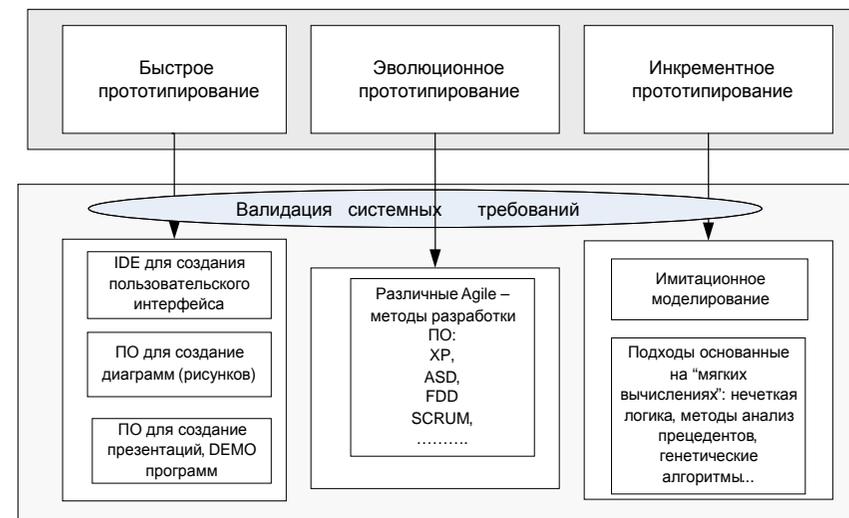


Рис. 1. Классификация моделей и средств прототипирования СТ

3. Концепция макро-технологии адаптивного проектирования. МТАП

может быть представлена в виде схемы адаптивного управления с обратной связью, включающей метрики и модели оценки эффективности такого управления [2]. Модели отображаются в соответствующие проекции многомерного информационного пространства, состоящего из четырех локальных пространств [3]: $P1$ - пространство моделей системных архитектур [2], $P2$ - пространство моделей системных требований, $P3$ - пространство методов проектирования, $P4$ - пространство моделей оценок и метрик. Все четыре пространства объединяются системной траекторией, которая задает конкретный вариант процесса адаптивной разработки КПП.

Пространство $P2$ строится на основе трех критериев оценки СТ, инвариантных по отношению к предметной области разрабатываемой ИУС. Этими критериями являются: 1) полнота спецификаций - определяет степень завершенности и полноты проектных спецификаций, 2) степень формализации - определяет, насколько формальным является представление СТ, 3) мера согласованности - определяет насколько согласованы между собой СТ с точки зрения различных специалистов, принимающих участие в их разработке (эксперты предметной области, аналитики, программисты и т.д.). Для анализа СТ был предложен эффективный метод оценки, основанный на аппарате нечеткой логики [9].

Пространство *П3* предназначено для принятия решений в области выбора проектного решения с учетом необходимых конструктивных параметров и имеющихся методов проектирования. Это пространство включает каталог шаблонов проектирования, необходимых для создания распределенных многоуровневых архитектур ИУС.

Пространство *П4* включает набор метрик, моделей и технологий для оценки атрибутов качества проектируемой ИУС, включая такие характеристики как производительность и надежность.

В соответствии с концепцией многомерного информационного пространства [3] МТАП реализует следующую последовательность операций в обозначенных пространствах. Появляющиеся системные конфликты представляются в виде нечеткого описания соответствующих СТ в пространстве *П2*. Далее при помощи разработанных методик определяется допустимая область, в рамках которой это требование выполняется. При помощи методов устранения нечеткости генерируется значение системного свойства, соответствующее поставленным требованиям. Затем при помощи одного из методов проектирования в пространстве *П3* строится набор допустимых проектных решений. Для каждого проектного решения производится процедура оценивания его атрибутов качества, включая производительность и надежность. Если результаты оценивания указывают на недопустимые значения атрибутов качества, то происходит возврат на предыдущий шаг в пространство *П3*, производится корректировка и построение новых проектных решений. Итеративный процесс продолжается до тех пор, пока не будет найдено соответствующее проектное решение. Когда такое решение найдено и согласованы другие параметры процесса разработки, включая финансовые ресурсы, график работ, решение документируется, шаг проектирования завершается и происходит переход на последующие шаги создания ИУС. Если требуемое проектное решение не может быть найдено, то происходит возврат в пространство *П2*, где выполняется корректировка СТ. На рисунке 2 показана графическая интерпретация этого процесса.

Далее мы более подробно рассмотрим локальные модели, разработанные для формализации процессов ИТЭП, которые итерационно протекают в пространствах *П3-П4* и тем самым обеспечивают получение адаптивных КПП.

4. Модели логических блоков интегрированной технологии эволюционного прототипирования (ИТЭП). С технологической точки зрения локальные пространства *П3-П4* представляются наборами соответствующих моделей, алгоритмов и инструментов. Так, например, пространство

П3 содержит ряд шаблонных архитектур для КПП в ИУС, при помощи которых создаются новые решения на основе существующих повторно используемых программных компонентов. На рисунке 3 показан пример шаблонной архитектуры КПП для построения межрегиональной распределенной ИУС реального времени [2]. На узле базового уровня, где происходит непосредственный обмен данными с Программируемыми Логическими Контроллерами (ПЛК) в режиме реального времени, ИУС содержит компоненты таких типов как Сервер Обработки Данных (СОД), Сервис Визуализации Данных (СВД), Сервис Архивирования Данных (СЗД), Сервис Синхронизации Данных (ССД), Интегрированная База Данных Узла (ИБДУ). СОД в свою очередь состоит из двух компонентов нижнего уровня – XProtocol и TechXObject. Перечисленные компоненты были реализованы с использованием технологий и программных решений Microsoft COM/DCOM, IIS, SQL Server, IE а также языков программирования и разметки данных C++, JavaScript, HTML [2].

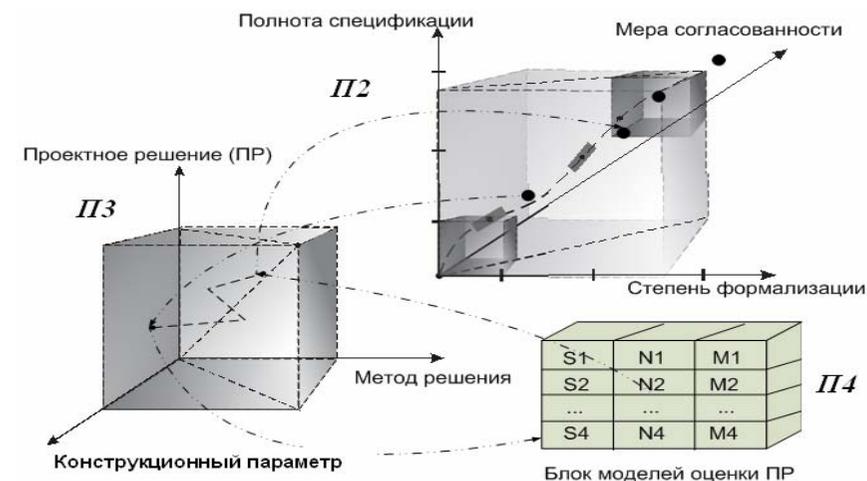


Рис. 2. Концептуальная модель технологии адаптивного проектирования

Пространство *П4* включает модели, алгоритмы и программные инструменты, предназначенные для оценивания атрибутов качества проектируемой ИУС. Этот инструментальный комплекс состоит из специализированной среды моделирования и моделей различного типа, которые могут быть выполнены в среде моделирования. Среда моделирования поддержи-

вает выполнение метрических и имитационных моделей, а также прототипов ИУС.

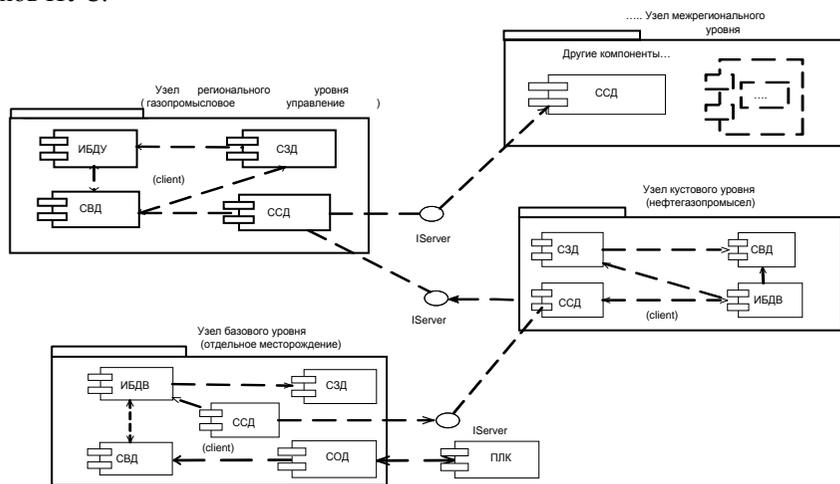


Рис. 3. Шаблонная архитектура распределенной ИУС реального времени

Имитационные модели играют значительную роль в процессах исследования отдельных атрибутов качества проектируемой ИУС [2,10]. Этот подход наилучшим образом подходит для моделирования сложных динамических систем, для которых построение адекватных аналитических математических моделей затруднено. Разработанный комплекс имитационного моделирования (КИМ) [10] является интегрированным программным пакетом, который включает в себя как реальные компоненты ИУС, которые исследуются, так и соответствующие имитационные модели с визуальной CASE-средой моделирования. КИМ позволяет исследовать свойства КПП и сделать вывод об удовлетворении предъявляемых к ним требований в условиях изменяющихся характеристик внешней среды ИУС. Помимо неоспоримых преимуществ имитационный подход имеет также ряд недостатков, основным из которых является то, что реализация имитационных моделей непосредственно не обеспечивает требуемой функциональности КПП, хотя по затратам может быть сопоставима с их непосредственной разработкой. В такой ситуации эффективным подходом является использование прототипирования искомых КПП [4].

ИТЭП обеспечивает возможность исследования атрибутов качества КПП, их нефункциональных свойств и сопоставления полученных результатов с предъявляемыми СТ. Для этого ИТЭП использует существующие

реальные компоненты, что позволяет эффективно использовать ресурсы, так как после прототипирования проектируемая ИУС в процессе разработки может эволюционировать непосредственно из готового прототипа. Основная идея ИТЭП состоит в отображении нефункциональных СТ в множество целевых свойств ИУС или ограничений. Структура прототипа создается на основе одного из архитектурных шаблонов, таким образом, чтобы выполнялись функциональные требования к ИУС. Затем неизвестные свойства прототипа исследуются в процессе его выполнения. Итерационный процесс прототипирования заканчивается, когда найдено решение со свойствами, удовлетворяющими поставленным СТ, после чего происходит переход к реализации ИУС на основе полученного прототипа. Структурно ИТЭП состоит из логических блоков PREPARE, BUILD, CONFIG, EXECUTE, ANALYSE, каждый из которых решает специфические задачи в ходе итеративного процесса прототипирования.

Блок PREPARE предназначен для получения области целевых свойств ИУС $TSP \subset PRSP$ для множества свойств $MRSP = \{(rsp)_i\}, i = \overline{1, |MRSP|}$ в пространстве на множестве свойств $PRSP = D_{(mrsp)_1} \times \dots \times D_{(mrsp)_{MRSP_1}}$ (где $D_{(mrsp)_i} = \{d_j\}, j = \overline{1, N_{(mrsp)_i}}$ - домен значений i -го свойства) на основе результатов экспертной обработки предъявляемых нефункциональных требований $MSRS = \{(srs)_i\}, i = \overline{1, |MSRS|}$. Обработка требований проводится при помощи стандартных процедур инженерии требований. Экспертная процедура обработки связана с тем, что зачастую СТ формируются в неформализованном виде, при помощи естественных языков, что вносит в спецификации СТ неоднозначность и зависимость от контекста.

Блок BUILD предназначен для выбора архитектуры прототипа. Каждый из компонентов, поддерживающих ИТЭП, обладает свойствами трех типов – статические, конфигурируемые, неизвестные. К статическим относятся такие неизменные свойства компонентов, которые зависят от внутренней реализации компонента и не могут быть изменены без изменения реализации компонента. Значения конфигурируемых свойств могут быть заданы на этапе настройки прототипа перед его выполнением и сохраняют свои значения в процессе выполнения и исследования прототипа. Неизвестные свойства – это те характеристики ИУС, которые должны быть найдены в процессе прототипирования, в том числе включающие искомые атрибуты качества исследуемой ИУС. Архитектура A , выбирается подмножества $MSTA$ множества имеющихся в репозитории шаблонов

ных архитектур МТА, $MTA = \{(mta)_i\}, i = 1..|MTA|$. MSTA – множество архитектур, статические свойства компонентов которых удовлетворяют поставленным требованиям $MSTA \subset MTA / MSTA^*$, где MSTA* - исследованные и отброшенные на предыдущих итерациях архитектуры. Процесс выбора одной архитектуры из доступного подмножества осуществляется при помощи методов принятия решений.

Блок CONFIG предназначен для задания значений конфигурируемых свойств компонентов архитектуры, найденной на предыдущем шаге. Значения конфигурируемых свойств могут определяться как допустимой областью значений, указанной в спецификациях требований, так и значениями свойств по умолчанию. $\forall k_i \in AK_r : \forall (mcp)_j \in MCP_{K_i} : (cp)_{k_i,j} \in CFS((mcp)_j)$, где AK_r - множество компонентов выбранной архитектуры, $CFS((mcp)_j)$ - процедура поиска значения $(cp)_j$ конфигурируемого свойства компонента $(mcp)_j$ по области допустимых значений TSP, $(cp)_j \in D_{(mcp)_j}$. Поскольку каждое свойство зачастую может иметь более одного допустимого значения, то для поиска окончательной конфигурации необходимо применение методов параметрического поиска.

После того, как прототип сконфигурирован, при помощи блока EXECUTE производится его установка в среду выполнения и запуск. Этот блок реализует ряд алгоритмов профилирования и сбора данных, необходимых для нахождения неизвестных свойств прототипа. Данные, собранные в процессе ряда запусков выполнения прототипа, имеют стохастический характер, поэтому их анализ и нахождение значений неизвестных свойств производится при помощи методов математической статистики.

Неизвестные свойства прототипа являются ключевыми в рассматриваемом итерационном процессе. Именно по значениям этих свойств определяется степень выполнения поставленных СТ и делается вывод и пригодности прототипа как основы для реализации ИУС. Если же прототип не удовлетворяет поставленным СТ, то происходит переход на один из предыдущих шагов и процесс прототипирования повторяется. Анализ и сравнение найденных характеристик производится в блоке ANALYSE. Решение принимается на основе критериев с использованием механизмов нечеткой логики. Ниже на рисунке 4 представлена общая блочная модель технологии.

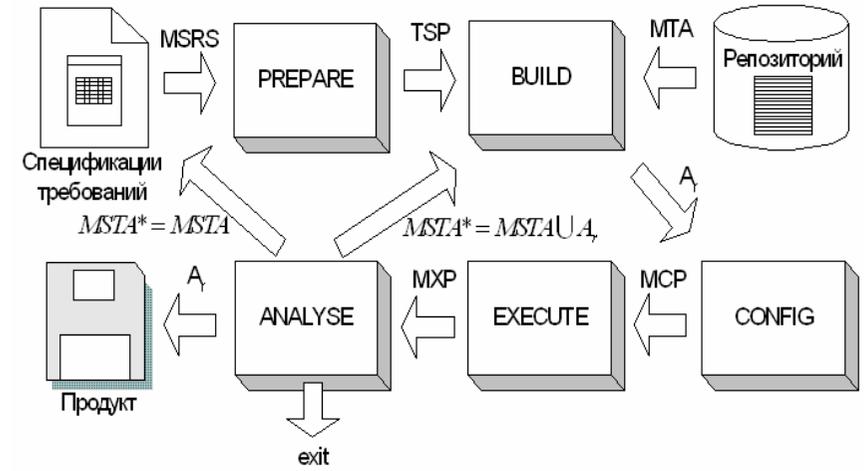


Рис. 4. Блочная модель ИТЭП

В качестве примера рассмотрим упрощенную модель процессов, происходящих в блоках EXECUTE и ANALYSE. Положим $U = \{u_i\}$ - множество неизвестных свойств, которые могут быть найдены в процессе исследования прототипа, что поддерживается как реализацией компонентов, так и настройками среды выполнения. $P = \{p_k\}$ - подмножество свойств выбранных для исследования в рамках определенной сессии прототипирования ($P \subset U$). В процессе выполнения прототипа накапливаются точечные значения v_i^k для каждого параметра k: $\forall p_k \in P : \exists V^k = \{v_i^k\}$. Точечные значения для всех профилируемых параметров могут быть заданы матрицей V:

$$V = \begin{bmatrix} v_1^1 & v_2^1 & \dots \\ v_1^2 & v_2^2 & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Для анализа данных и нахождения неизвестных свойств имеется множество методов анализа $M = \{m_i\}$. Матрица $MP = \{mp_k^i\}$ определяет, какие методы анализа могут применены к каждому из неизвестных параметров, что зависит от природы свойства, вида собранных данных, реализации самого метода. mp_k^i будет иметь значение 1 если метод k применим

для параметра i и 0 в противном случае. После вычислений мы получим результирующую матрицу $R = \{r_k^i\}$, где r_k^i - результат вычислений для свойства i с применением метода k : $r_k^i = F(V^k, mp_k^i, m_i)$.

Выводы. В результате проведенных исследований показаны методологические основы необходимости и целесообразности применения ИТЭП в общей схеме МТАП сложных ИУС, создаваемых с использованием КНР. Представлены также разработанные модели, позволяющие формализовать все основные этапы ИТЭП. Научная новизна данного подхода заключается в применении современных методов прототипирования к области исследования атрибутов качества ПО. Практическая значимость результатов состоит в повышении эффективности процессов создания ИУС путем сокращения затрат ресурсов и повышения качества получаемых проектных решений. Перспективной дальнейших исследований в этом направлении является совершенствование полученных моделей и разработка методики многокритериального сравнения эффективности различных способов обеспечения атрибутов качества КНР.

Список литературы: 1. Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. – СПб.: Питер, 2005. – 412 с. 2. Архитектуры, модели и технологии программного обеспечения информационно-управляющих систем / Н.В. Ткачук, В.А. Шеховцов, Д.В. Кукленко, В.Е. Сокол / Под ред. М.Д. Годлевского. – Харьков: НТУ «ХПИ», 2005. – 546 с. 3. Ткачук Н.В., Годлевский М.Д. Адаптивная технология разработки и реинжиниринга сложных ИУС в среде многомерного информационного метaprостранства // Труды 7-й международной научно-практической конференции «Современные информационные и электронные технологии», 22-26 мая 2006 г., г. Одесса. - Т.1 - С. 13. 4. Ткачук Н.В., Земляной А.А., Гамзаев Р.А. Технология эволюционного прототипирования компонентных программных решений для информационно-управляющих систем. // Тезисы докладов IX международной научно-технической конференции «Системный анализ и информационные технологии», НТУУ «КПИ», Киев, 15-19 мая 2007 г. - с. 206. 5. Luqi, R., Steigerwald, G., Hughes, V., Berzin A. CAPS as a Requirements Engineering Tool.. Proc. Conference on TRI-Ada, San Jose, CA, 75-83, 1991. 6. Herranz, A, Moreno-Navarro, J. Rapid Prototyping and Incremental Evolution Using SLAM, Proc. 14th IEEE International Workshop on Rapid Systems Prototyping, San Diego, 201-209, 2003. 7. Axure Software Solutions, <http://www.axure.com/products.aspx>. 8. iRise Company, <http://www.irise.com/>. 9. Ткачук Н.В., Гамзаев Р.А. Нечеткий подход к решению задачи анализа системных требований в процессах разработки и сопровождения информационных систем // Автоматизированные системы управления и приборы автоматизации. – Харьков: ХНУРЭ. – 2006. - Вып. 134. - С. 64- 70. 10. Ткачук Н.В., Горелый А.В., Земляной А.А. Комплекс имитационных моделей для исследования компонентных программных решений в ИУС АСУ ТП // Вісник Національного технічного університету "ХПИ". - Харків: НТУ "ХПИ". – 2004. - № 18, – С.145-152.

Поступила в редакцию 24.02.08

УДК 004.4'22

В.В. ЛАНИН, Пермский государственный университет,
г. Пермь, Россия

Л.Н. ЛЯДОВА, канд. физ.-мат. наук, АНО науки и образования
“Институт компьютеринга”, г. Пермь, Россия

ИНТЕЛЛЕКТУАЛЬНЫЕ CASE-СРЕДСТВА ДЛЯ РАЗРАБОТКИ АДАПТИРУЕМЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Представлено підхід до створення інтелектуальних інформаційних систем, що є основою CASE-технології METAS, призначеної для підтримки життєвого циклу динамічно адаптуємих розподілених інформаційних систем. Системи, створені за допомогою CASE-засобів METAS, функціонують у режимі інтерпретації. Вони базуються на багаторівневих мета-даних, які описують предметну область системи. Можливості адаптації реалізуються через засоби реструктуризації даних, налаштування користувальницького інтерфейсу, генерації запитів і звітів, динамічного підключення програмних компонентів сторонніх розроблювачів. Одна з найбільш важливих - підсистема керування документами, що забезпечує можливості інтелектуального пошуку, аналізу автоматичної класифікації й каталогізації документів, заснована на використанні онтологій й агентного підходу.

Представлен подход к созданию интеллектуальных информационных систем, являющийся основой CASE-технологии METAS, предназначенной для поддержания жизненного цикла динамически адаптируемых распределенных информационных систем. Системы, созданные с помощью CASE-средств METAS, функционируют в режиме интерпретации. Они базируются на многоуровневых метаданных, которые описывают предметную область системы. Возможности адаптации реализуются через средства реструктуризации данных, настройки пользовательского интерфейса, генерации запросов и отчетов, динамического подключения программных компонентов сторонних разработчиков. Одна из наиболее важных – подсистема управления документами, обеспечивающая возможности интеллектуального поиска, анализа автоматической классификации и каталогизации документов, основанная на использовании онтологий и агентного подхода.

An approach to development of intelligent information systems is presented. This approach is the base of CASE-technology METAS intended for support of adaptable distributed information systems life cycle. Information systems created with METAS CASE-tools operates in interpretation mode. It's based on multilayer metadata which describes universe of discourse for the information system. The adaptation capabilities are realized by means of data restructuring tools, interface generation and tuning, query builder and reports generation, dynamic linking of program components developed by exterior application designer. Document management subsystem is one of the most important. It includes tools of storing documents in database, business intelligent means, automatic classification and intelligent search of electronic documents. This subsystem is based on ontologies and multi-agents architecture.

Введение. Адаптируемость является важнейшим свойством современных информационных систем (ИС), которое проявляется как нефункциональ-