

References

1. **V P Klimanov** Methods of developing analytical models for the analysis of local area networks used in the management process. A training manual for the course. Edited by A.P. Ereemeev, MPEI, Moscow 1995. 114p.
2. **V P Klimanov, K V Diomin** Multiplex LAN data exchange «Three-dimensional square lattice» and its analytical model. Reports of an international conference «Information tools and technologies». В 1. -М., 1997.с. 84-91.
3. **V Klimanov, K Diomin** Multichannel Local Network «Three - Demensional Lattice» And Its Analytical Model. In book 43 Internationales wissenschaftliches kolloquium. Band 1. : Technische universitat ilmenau thuringen, 1998.p.84-91.
4. **V A Zhozhikashvili, V M Vishnevsky** Network of mass service. The theory and application of computer networks. -М.: Radio & Communications, 1988. -192 p.
5. **A Ferrari.** Assessment performance computing systems. -М.: MIR, 1981. -576 с.
6. **V A Bystrikova, V P Klimanov** Using the criterion of survivability in the analysis of networks with the topology «lattice». International Informatization Forum - 2001: Reports of an international conference «Information Systems and Technology» 16-18 October 2001, at 3 TT T3-М.: Edition of «Stankin», 2001.-134 p., p. 87-90.
7. **V P Klimanov, M V Sutyagin** Ensuring the reliability of information systems using network clustering. Magazine Journal of Computer and Information Technology, № 4, 2007, F.: Mechanical Engineering.
8. Cluster high availability solutions. <http://www lynx.ru/unix/ha.html>.

Климанов Вячеслав Петрович - заместитель заведующего кафедрой «Информационные системы» МГТУ «Станкин», докт. техн. наук, профессор, академик, вице-президент отделения «Информационные средства и технологии» Международной Академии Информатизации

*Klimanov V P - deputy head of the department «Information Systems» MSTU «Stankin», Ph.D., professor, academician, vice-president office «Information tools and technologies» International Academy of Informatization
E-mail: slav@stankin.ru*

УДК 62-523:681.3.067

Н.В. Козак, Р.А. Абдуллаев
N V Kozak, R A Abdullaev

Динамическое конфигурирование прикладных программных компонентов в приложениях электроавтоматики

Dynamic configuration process of applied components of PLC control systems

Изложены принципы управления динамической загрузкой и выгрузкой компонентов в прикладных приложениях электроавтоматики. Исследовано место в архитектуре и принципы работы ответственного за этот процесс менеджера компонентов. Определены программные интерфейсы для построения компонентов динамического конфигурирования на примере CoDeSys.

For PLC control systems applied program components the principles of dynamical configuration process are given. The manager, which controls this process, was described. For manager of dynamical configuration the place in architecture, the mechanisms of work and common interfaces are investigated. The CoDeSys PLC software was analyzed as the base for architecture.

Ключевые слова: программное обеспечение, система управления, электроавтоматика, CoDeSys, конфигурирование программных компонентов, плагин, компонентная архитектура, менеджер компонентов, динамическая выгрузка компонентов

Keywords: software, control system, PLC, CoDeSys, configuration of program components, plug-in, module architecture, components manager, dynamical configuration of components

Современный пользовательский интерфейс должен обладать конкурентоспособными характеристиками привлекающими эргономичностью и удобством в использовании. В определённой мере это справедливо и для приложений систем управления. Ведущие разработчики систем имеют возможность уделять внимание (средства и ресурсы) данной проблематике [1].

Одним из перспективных направлений для решения задач построения пользовательского интерфейса в системах управления является использование готовых решений, которые успели зарекомендовать себя и понравиться пользователям [2]. В качестве примера можно представить использование графических редакторов управляющих программ электроавтоматики производства 3S в системах TwinCAT и IndraWorks (в подсистеме IndraLogic).

Графические элементы, логика их моделей, внешние загружаемые ресурсы компонентов пользовательского интерфейса могут занимать не соизмеримо большее пространство оперативной памяти, выделяемой для работы приложений системы управления, чем прочие даже более важные процессы. С этой точки зрения возникает необходимость выгружать избыточные для режимов работы системы компоненты. В первом приближении, осуществление задачи динамической выгрузки и загрузки, исполняемых в рамках процессов систем управления специальных съёмных программных компонентов (плагинов), позволит реализовать следующие преимущества:

- снижать нагрузку на вычислительные ресурсы аппаратных средств;
- как следствие первого, повышать быстродействие исполнения операций пользователя;
- снижать вероятность возникновения ошибок из-за некорректной работы не востребованных модулей.

Целью статьи является определение принципов конфигурирования загрузкой и выгрузкой программных компонентов в приложениях систем управления электроавтоматикой с открытой архитектурой и последующая реализация на основе предложенной теории. Для достижения цели необходимо решить следующие задачи:

1. Проанализировать механизмы динамической загрузки и выгрузки программных компонентов, предоставляемый платформой .NET;
2. Специфицировать необходимые программные интерфейсы для построения механизма динамического конфигурирования;
3. Реализовать возможности предложенного механизма в разработке средств динамического конфигурирования в системе управления электроавтоматикой CoDeSys.

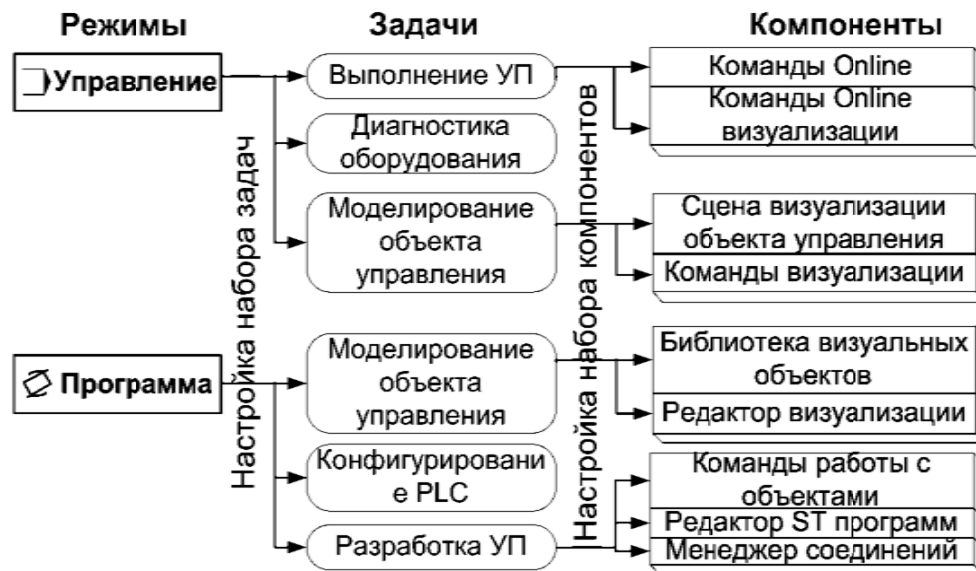


Рис. 1. Конфигурация компонентов системы управления для двух режимов: Управление и Программа

На схеме (рис. 1) показана общая структура режимов работы системы и динамического использования программных компонентов, осуществляющих решение выбранных задач в определённых режимах работы. Из схемы видно, что для решения той или иной задачи требуется несколько компонентов, каждый из которых требует определенных затрат вычислительных ресурсов. Для их экономии необходимо осуществлять взаимоисключающую загрузку и выгрузку съёмных компонентов (плагинов), отвечающих за решение поставленной задачи.

Современным решением конфигурирования программных компонентов, загруженных в систему управления электроавтоматикой, является применение платформы .NET. Отдельный процесс .NET может содержать в себе один или несколько доменов приложения, который в свою очередь может содержать в себе несколько потоков. Домен приложения предоставляет функциональность загрузки и выгрузки сборок, которые содержат в себе компоненты, загружаемые в систему. Анализ платформы .NET показал, что с помощью доменов приложения можно реализовать задачу динамического конфигурирования компонентов системы управления электроавтоматикой.

Для примера построения средств динамического конфигурирования компонентов в системе CoDeSys [3], необходимо воспользоваться функциональностью класса ComponentManager и расширить его возможности путем добавления новых методов и (или) переопределения уже реализованных. Данный подход показан на схеме (рис. 2). Класс ComponentManagerExt наследует функции базового класса ComponentManager и дополняет их возможностью динамического конфигурирования компонентов. Для осуществления этого подхода необходимо выполнить дизассемблирование открытого системного плагина Core. Этот процесс осуществляется при помощи утилиты Reflector.exe. В результате этой операции будет получен исходный код, который позволит реализовать производный класс ComponentManagerExt. Этот класс будет располагаться в компонентах каркаса системы CoreDynLoad.

Новый программный компонент для системы CoDeSys будет специфицировать интерфейс IDynamicPlugin, в котором содержатся функции, необходимые для загрузки и выгрузки сборок из системы. Если ключевые классы, находящиеся в плагине, будут поддерживать этот интерфейс, то при смене режима работы будет происходить динамическое перераспределение таких компонентов для решения текущих задач. В противном случае переконфигурирование осуществляться не будет. Следовательно, компонент находится постоянно в памяти программы и отнимает вычислительные ресурсы.

Утилита конфигурирования режимов IPM_Ex имеет функционал, предоставляющий пользователю возможность выбирать загружаемые в систему плагины. С ее помощью настраивается соответствие между наборами режимов, задач и компонентов (рис. 1). Эта информация сохраняется в xml-формате в файл конфигурирования режимов. При запуске программы создается системный объект класса ComponentManagerExt, который выполняет загрузку настроек конфигурации режимов из xml-файла. После этого во время работы приложения система управления сможет производить динамическое перераспределение компонентов для актуальных задач. Для такого перераспределения будет служить плагин RegimsConf. В нём при переходе системы в очередной режим будут производиться запросы к новому менеджеру компонентов.

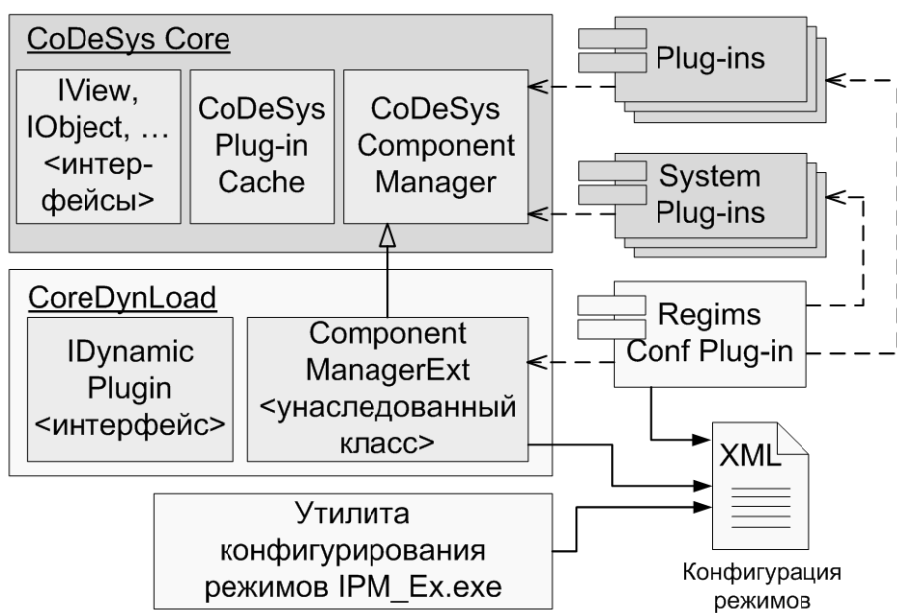


Рис. 2. Принцип работы системного плагина конфигурирования компонентов

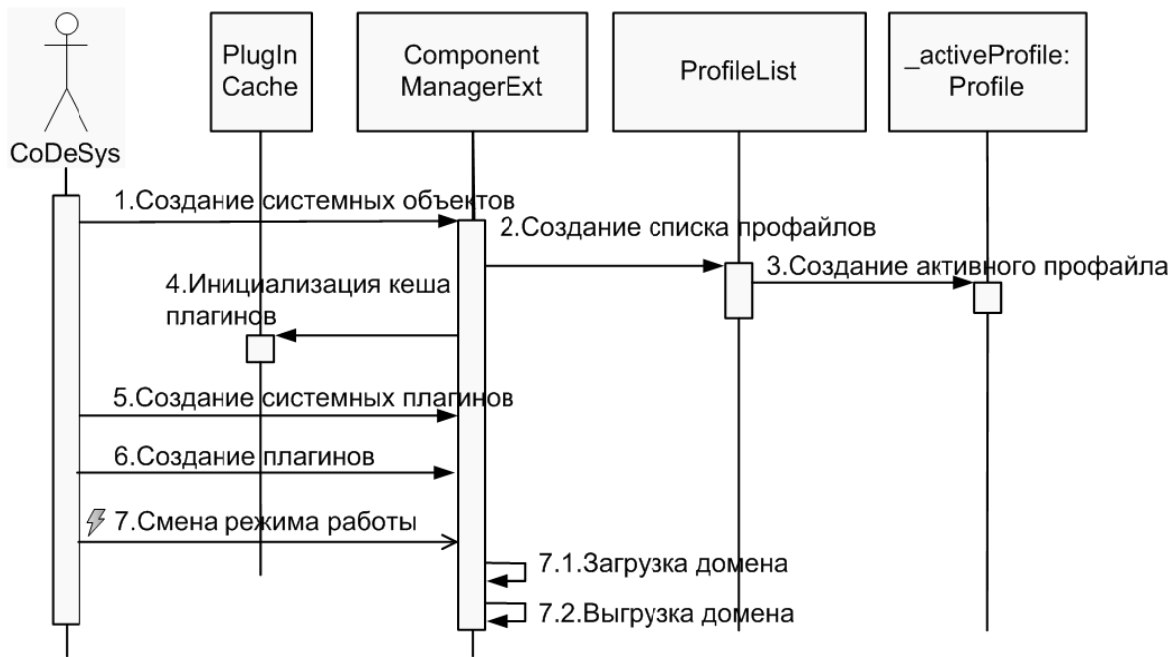


Рис. 3. Последовательность работы расширенного менеджера компонентов

Схематично последовательность работы менеджера компонентов ComponentManagerExt представлена на диаграмме последовательности (рис. 3). Менеджер компонентов создается один раз при запуске системы (вызов 1). В своем конструкторе менеджер создает список файлов возможных конфигураций загрузки компонентов (профайлов) (2). После чего на основе активного профайла (вызов 3) происходит инициализация кэша плагинов (вызов 4). Теперь менеджер компонентов готов к созданию плагинов (вызов 6), в том числе и системных (вызов 5). При смене режима (вызов 7) работы системы менеджер компонентов загружает (вызов 7.1) и выгружает (вызов 7.2) соответствующие домены.

На программном уровне задачи, решаемые системой управления, объединены в домены приложений. Каждый домен приложения содержит в себе совокупность компонентов, необходимых для работы системы в определенном режиме. Менеджер компонентов содержит в себе столько доменов приложений, сколько задач решает система. При этом в каждом режиме загружены только те домены, которые необходимы для решения текущих задач. При смене режима работы системы происходит загрузка требуемых доменов, и выгрузка тех доменов, которые стали не актуальны на данном этапе исполнения (Рис. 4).

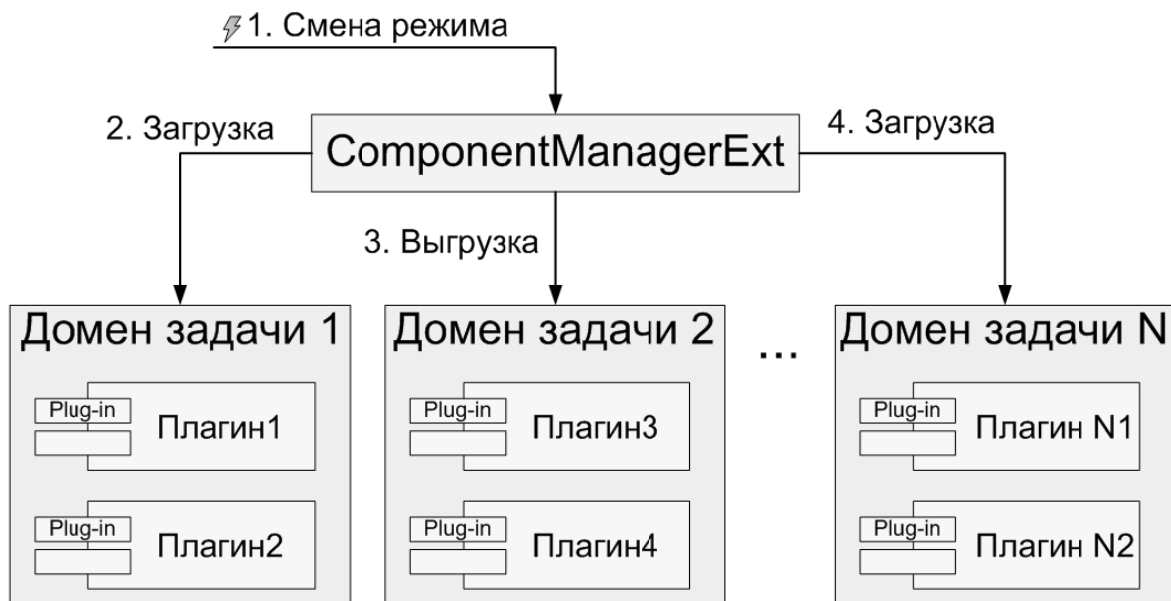


Рис. 4. Работа с доменами менеджера компонентов

Результатом решения поставленных задач статьи является то, что пользователь получает приложение, которое адекватно и своевременно реагирует на изменения в текущей конфигурации режима системы. Достижение решения поставленной задачи позволит экономить системные ресурсы при работе в различных режимах. В частности для текущего режима работы системы будут исключены затраты на функционирование не востребованных модулей. За счет выгрузки компонентов система будет застрахована, во-первых, от излишнего выделения вычислительных ресурсов, приводящих к снижению производительности системы, во-вторых, от возможных сбоев которые могли бы происходить в выгруженных модулях. После решения задачи построения системы динамического конфигурирования компонентами подобные проблемы будут исключены.

Библиографический список

1. **Мartiнов Г.М., Козак Н.В.** Декомпозиция и синтез программных компонентов электроавтоматики // Приборы и системы. Управление, контроль и диагностика. 2006. №12. С. 4-11.

2. **Козак Н.В.** Основы интеграции компонентов электроавтоматики на примере графического редактора 3D сцены визуализации //

Труды Международной научно-технической конференции «Информационные средства и технологии». Москва 2007. – Том 3 – С. 132 – 135.

3. **Петров И.В.** Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / Под ред. проф. В. П. Дьяконова. – М.: СОЛОН-Пресс, 2004. 256 с. ISBN 5-98003-079-4

References

1. **G M Martinov, N V Kozak** Decomposition and synthesis of the logic systems software components. The Instruments and systems. Management, checking and diagnostics. 2006. №12. P. 4-11.

2. **N V Kozak** The base conception of PLS components integration by the example of 3D scene

graphical editor // Works to International research conference “Information facilities and technologies”. Moscow 2007. That 3 – P. 132-135.

3. **I V Petrov** The Programmed controllers. Standard languages and receiving the applied designing. Moscow, SOLON-Press, 2004. 256 pages, ISBN 5-98003-079-4

Козак Николай Владимирович - преподаватель кафедры "Компьютерные системы управления" МГТУ «Станкин», магистр.

E-mail: kozak@ncsystems.ru

Абдуллаев Роман Ахматалиевич - магистрант кафедры "Компьютерные системы управления" МГТУ «Станкин».

E-mail: abdullaev@ncsystems.ru

Kozak N V - teacher of department "Computer-architecture control systems" MSTU «Stankin», master of Science.

E-mail: kozak@ncsystems.ru

Abdullaev R A - graduate student of the MSTU «Stankin».

E-mail: abdullaev@ncsystems.ru