Московский Государственный Технический Университет им. Н. Э. Баумана

Дипломная работа

Многоагентная автоматизированная система контроля обновлений программного обеспечения в реальном масштабе времени

Панфиленко С. А.
ИУ4-Д2, 2007 г.

Научный руководитель: доцент, к. т. н. Власов А. И.

Цели и задачи работы

Цель работы:

Исследование существующих структур обновлений и способов автоматизации тестовых сценариев, разработка системы автоматизированного контроля выкладки обновлений на публичные сервера.

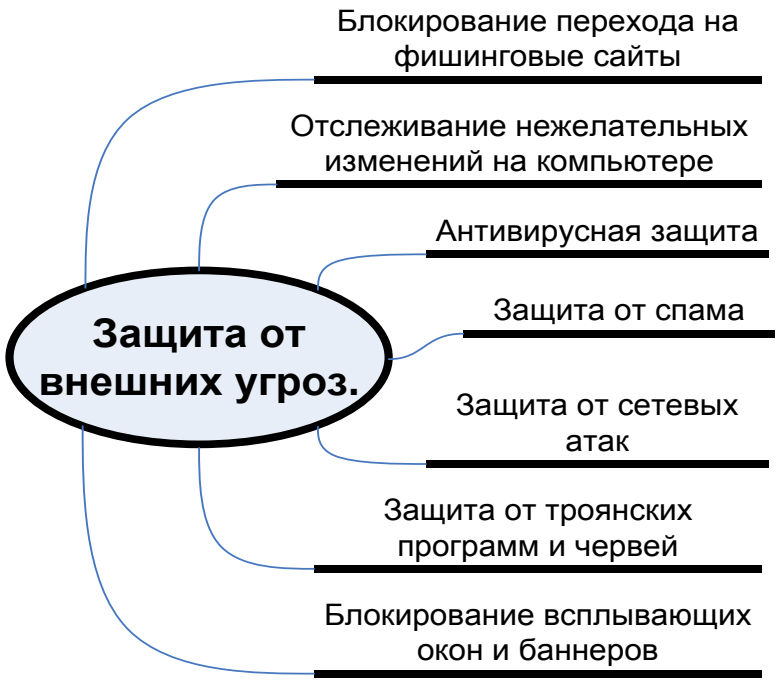
Решаемые задачи:

- Анализ структуры обновлений
- Анализ СПО и необходимой инфраструктуры
- Разработка сценариев контроля обновлений
- Сравнительный анализ способов автоматизации сценариев контроля обновлений
- Разработка архитектуры автоматизированной системы
- Разработка методов управления агентами
- Разработка модулей системы
- Экспериментальная оценка работоспособности системы

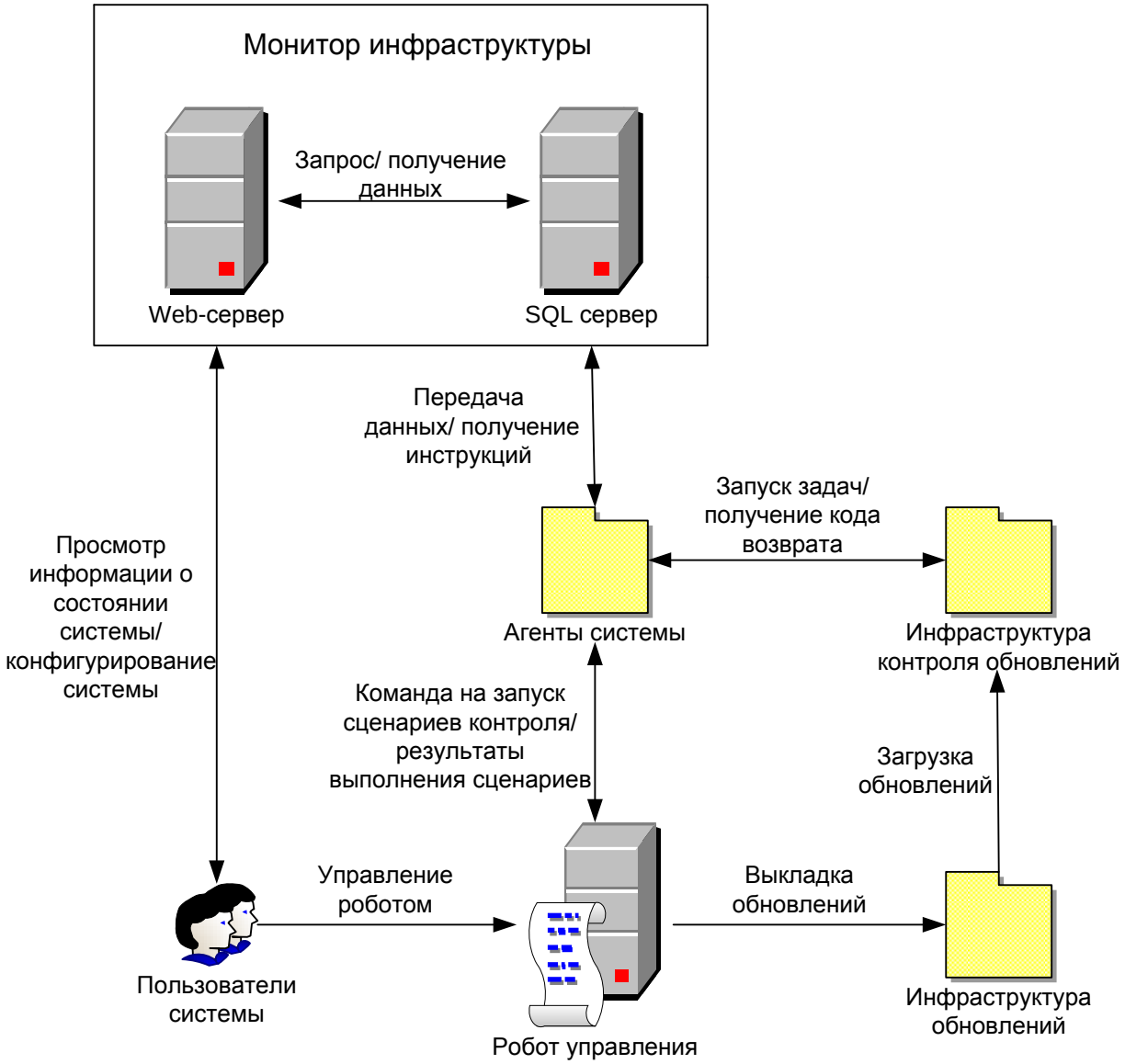
Существующие проблемы

- Необходимость своевременной реакции на новые виды угроз (распознание, поиск метода предотвращения, выпуск в виде обновления)
- Большое количество СПО, обеспечивающее защиту от внешних угроз
- Размещение всех обновлений на одном ресурсе
- Идентификация приложением обновлений, предназначенных именно этому приложению
- Изменение со временем структуры обновлений и необходимость поддержания старых структур
- Большая частота выкладки обновлений на публичные сервера (каждый час)
- Большие объемы вирусной (0.5 Тб) и чистой (несколько Тб) коллекций

Разновидности защиты от внешних угроз



Архитектура построения автоматизированной системы

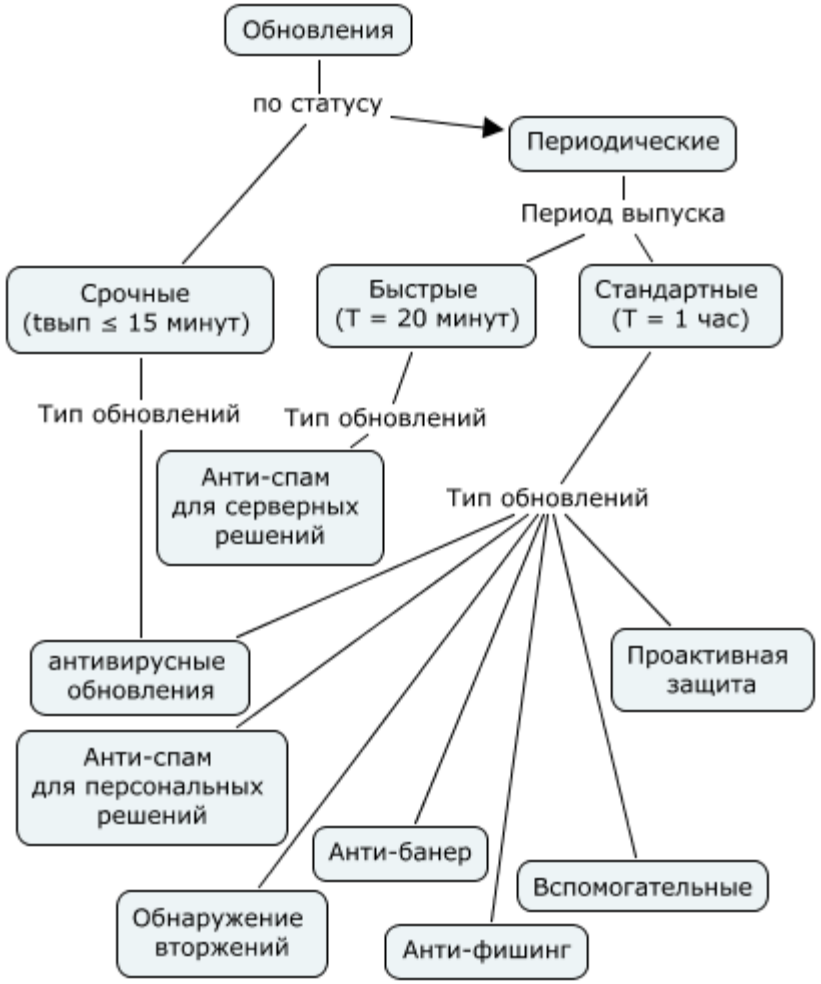


Классификация обновлений СПО

по назначению:

- антивирусные обновления
- антивирусные обновления потокового режима компонента Веб-Антивирус
- обновления компонента Анти-Спам
- обновления сетевых атак – система обнаружения вторжений
- обновления компонента Анти-Баннер
- обновления компонента Анти-Фишинг
- обновления компонента Проактивная защита: анализ активности приложений, мониторинг реестра
- файлы описания списка доступных компонент для задачи копирования обновлений
- файл заблокированных ключевых файлов black.lst

по статусу:



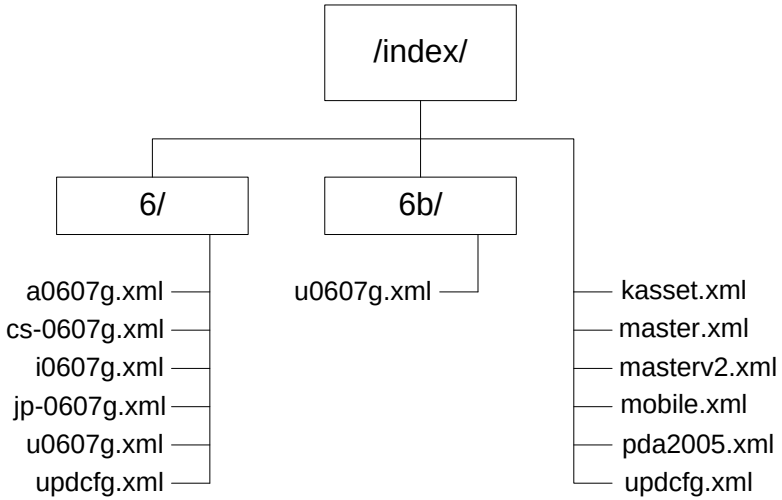
Файл-индекс – файл описания списка доступных компонент для задачи копирования обновлений, содержащий также инструкции по выполнению задачи обновления и требования к корректному применению продуктом новых обновлений.

Мастер-индекс – файл описания структуры обновлений (схемы индексов) для данного СПО. Содержит необходимые ссылки на вспомогательные файлы-индексы.

Существующие схемы индексов

- Master.xml («пятерочная»)
- i0607g.xml/a0607g.xml («простыни»)
- Masterv2.xml («костыль»)
- U0607g.xml («шестерочная»)

Граф-классификация мастер-индексов



Описание индексов

- kasset.xml - схема индексов модуля защиты от спама для старых версий СПО
- mobile.xml/ pda2005.xml - схемы индексов для мобильных устройств
- updcfg.xml - индекс переченя серверов обновлений
- cs-0607g.xml/ jp-0607g.xml - схемы индексов для кастомизированных продуктов

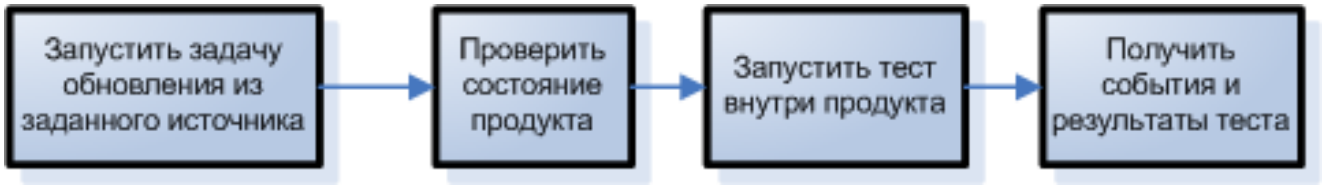
Критерии включения СПО в систему контроля

1	Дата выпуска продукта не ниже 01.01.2006 (максимальный период технической поддержки продукта – 2 года, предполагаемая дата начала функционирования системы в полном объёме – 01.01.2008г.)
2	Исключить продукты для серверных решений (кроме Kaspersky Anti-Virus 6 for Windows File Servers и Kaspersky Administration Kit)
3	Исключить продукты для мобильных устройств
4	Исключить продукты для Linux платформ по причине отсутствия разработчика под данный вид инфраструктуры
5	Исключить продукты, которые полностью входят в состав многомодульных продуктов, включенных в систему контроля

Требования к мониторингу системы

1	системная информация (CPU/RAM – загрузка процессора и памяти, HDD – использование жестких дисков)
2	доступность сетевых ресурсов на чтение/ запись
3	доступность серверов (ftp, http, smtp)
4	доступность и работоспособность роботов
5	доступность и работоспособность сервиса ЭЦП
6	работоспособность сервиса репликации
7	работоспособность сервиса перевыпуска обновлений
8	работоспособность дублирующей системы мониторинга
9	температура в серверной
10	события об изменении статуса компонент (перезагрузка, выключение, обновление версии, выполнение тестовых сценариев)

Общая схема проверки обновлений



Выбранные способы для анализа

- KAVCON – утилита проверки антивирусных баз, используемая на данный момент в инфраструктуре выкладки и проверки обновлений
- JTS Safe Protector – реализация сканирования файлов на основе СПО шестой версии (сборка 411) без графического интерфейса по средствам командной строки
- СПО Kaspersky Anti-Virus 6.0 MP1 (сборка 411)
- СПО Kaspersky Anti-Virus 6.0 MP2 (сборка 621)

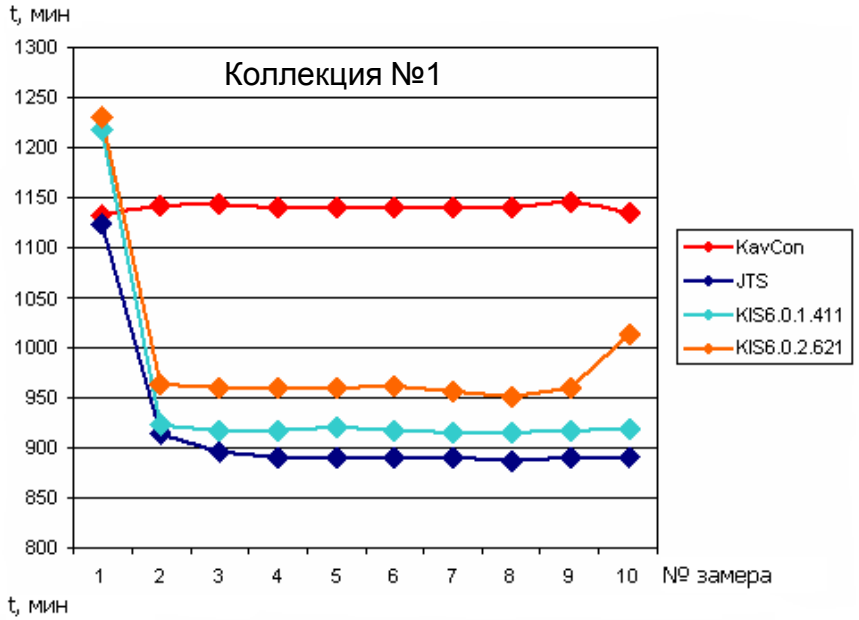
Созданные тестовые коллекции для анализа

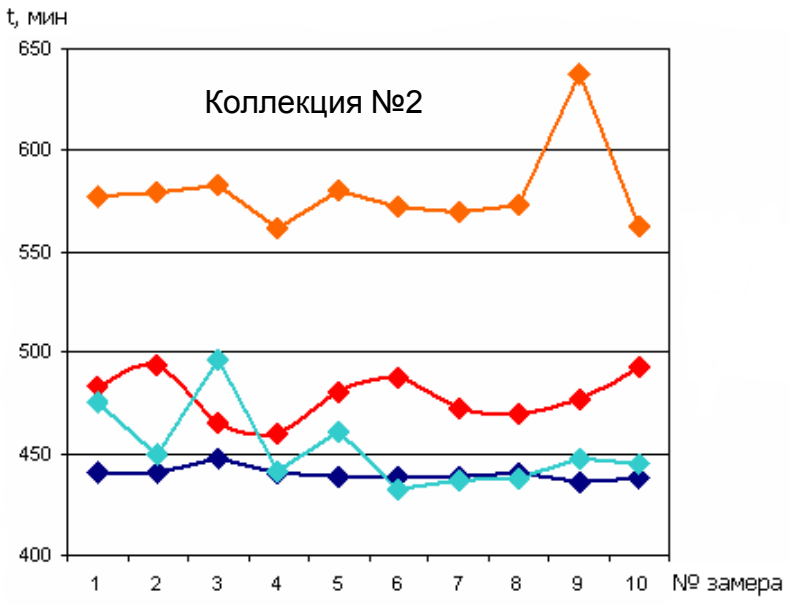
- белая коллекция №1 – состоит из следующих файлов: Windows, Program Files, Documents and Settings (Windows XP SP2 сразу после установки) + развёрнутые установки MS Office 2003 и Adobe Acrobat 7.0 Professional
- вирусная коллекция №2 – все файлы взяты из разных источников
- вирусная коллекция №3 – часть реальной вирусной коллекции класса TrojWare

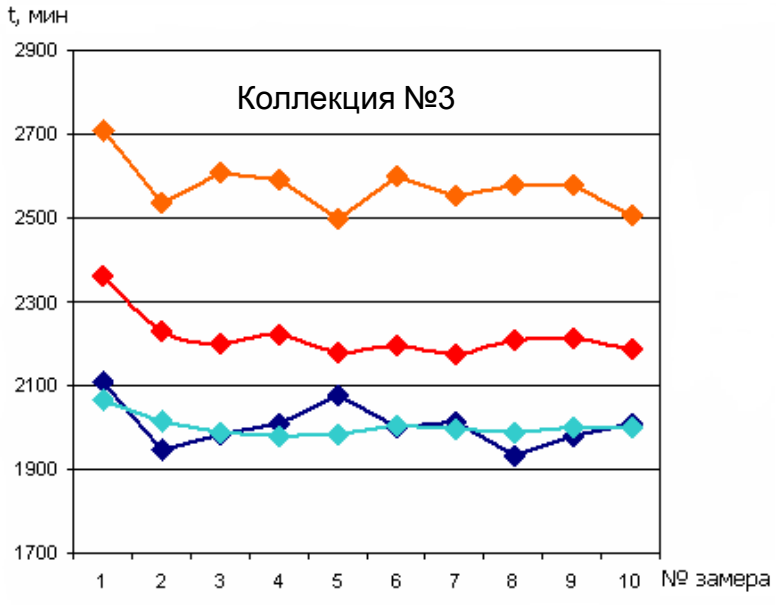
Варианты анализа

- последовательное сканирование файлов
- перезагрузка рабочей станции перед каждым сканированием файлов

Анализ способов проверки антивирусных баз



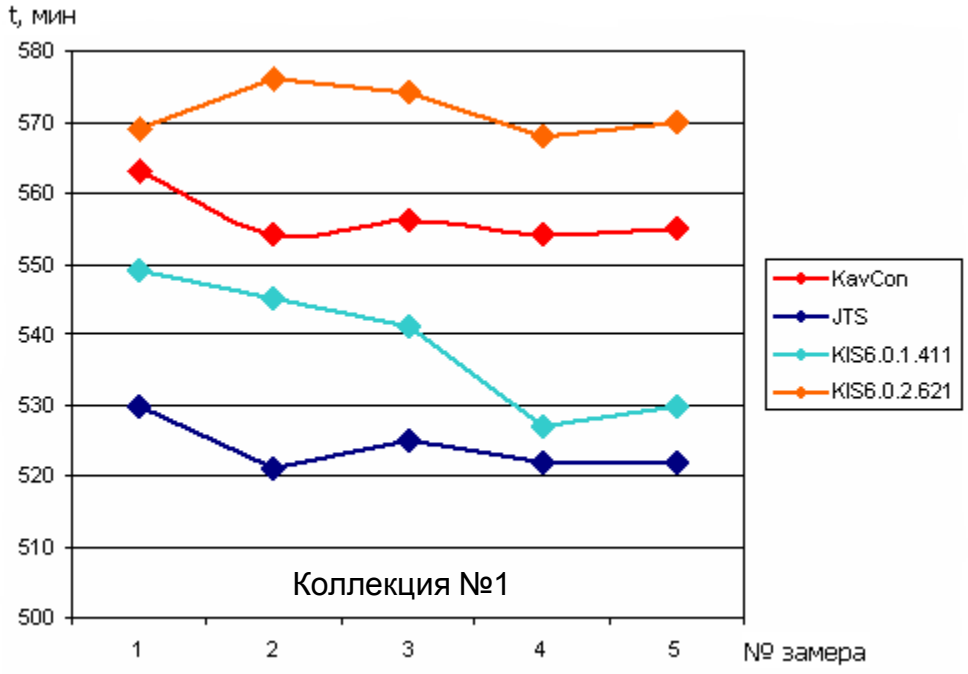


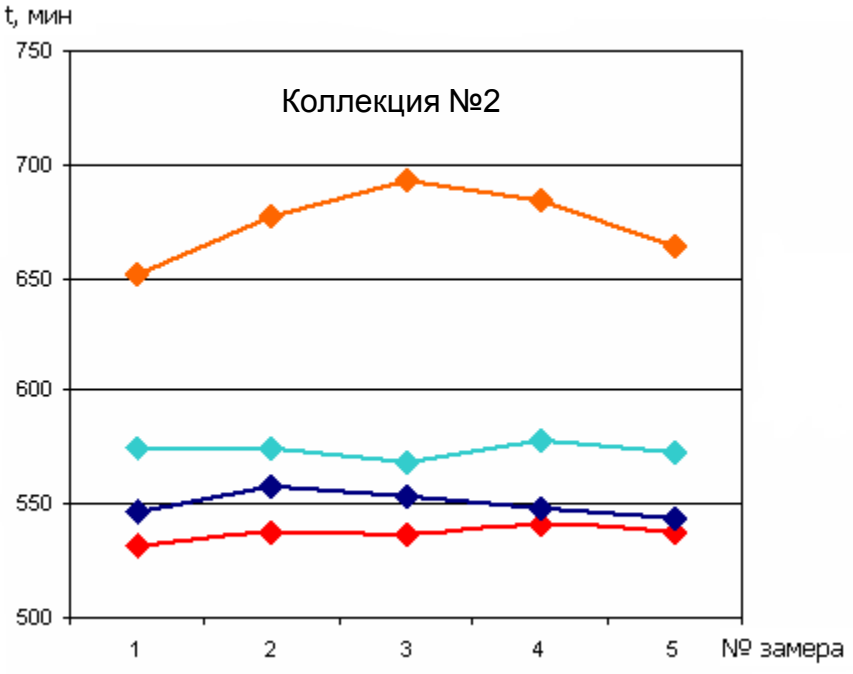


Выводы

- значительное отставание KAVCON от продукта при сканировании файлов, не являющихся вирусами.
- существенная зависимость скорости сканирования продуктом файлов от кэш
- деградация скорости сканирования заражённых файлов при использовании последней версией СПО
- JTS не на много, но быстрее KAVCON'а производит сканирование зараженных файлов

Анализ способов проверки антивирусных баз

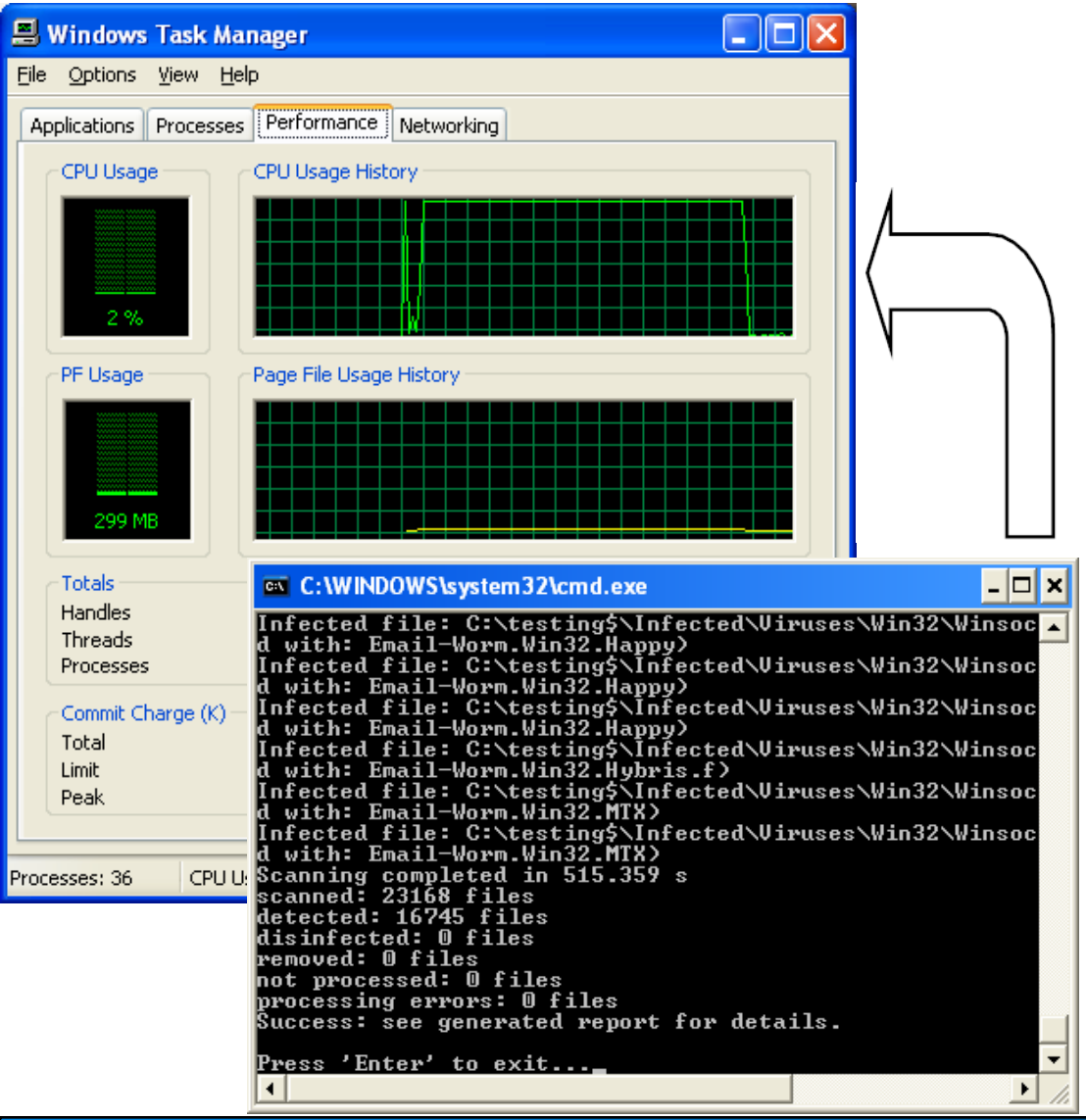




Выводы

- существенная зависимость последней версии СПО от кэш
- преимущество в скорости сканирования чистых файлов JTS над KAVCON'ом
- KAVCON обрабатывает зараженные файлы немного быстрее, чем JTS, без использования кэш
- KAVCON не актуален на сегодняшний день и необходима замена данного метода проверки обновлений на метод JTS.

Загрузка памяти при сканировании вирусной коллекции файлов

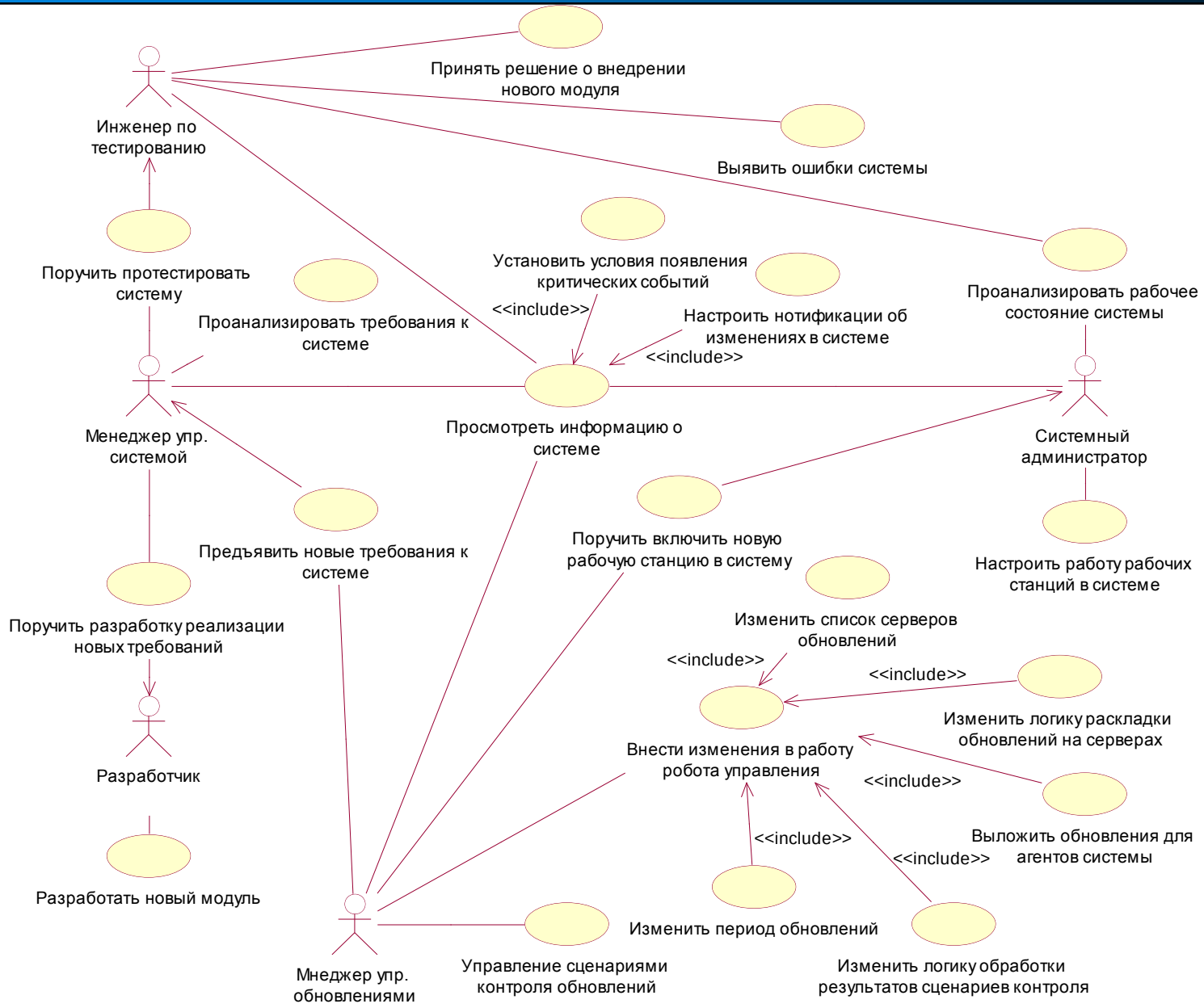


Одной из проблем при контроле обновлений является большой объем файловых коллекций для проверки антивирусных обновлений.

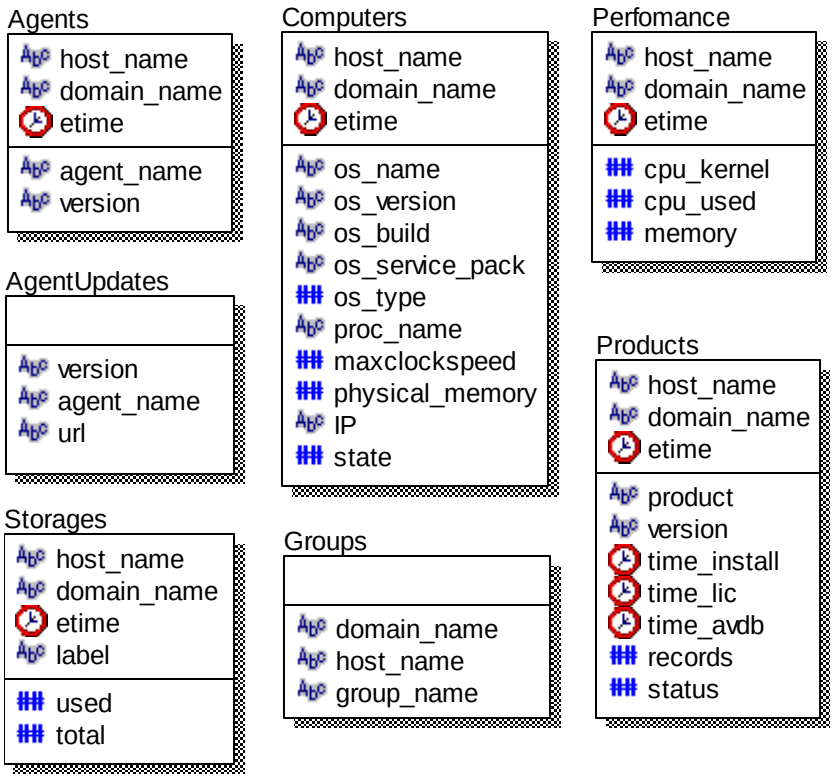
СПО, применяющее антивирусные базы при сканировании файлов, при всех прочих условиях осуществляет 100% загрузку памяти рабочей станции, что приводит к рассеиванию максимального значения мощности процессора, а, следовательно, повышению температуры процессора.

Повышение температуры является одной из причин снижения производительности, что недопустимо в условиях жестких временных рамок на проведение контроля обновлений.

Разработка вариантов использования системы



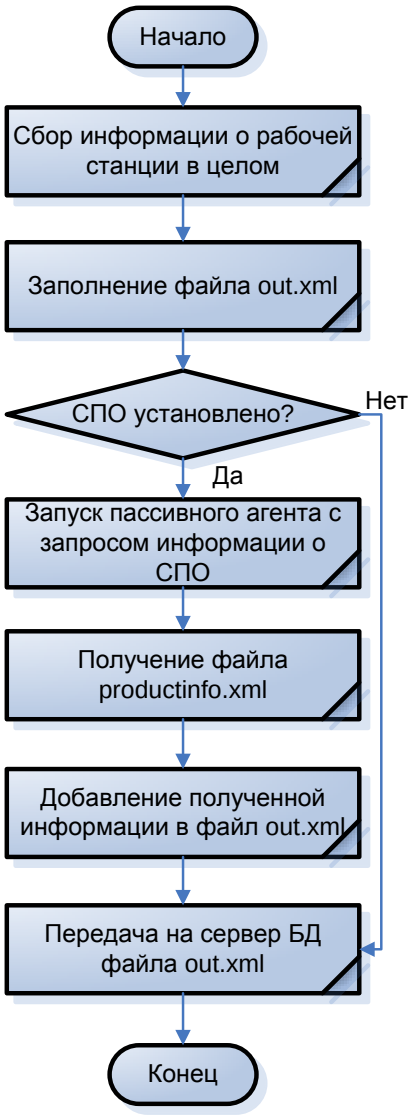
Логическая информационная модель



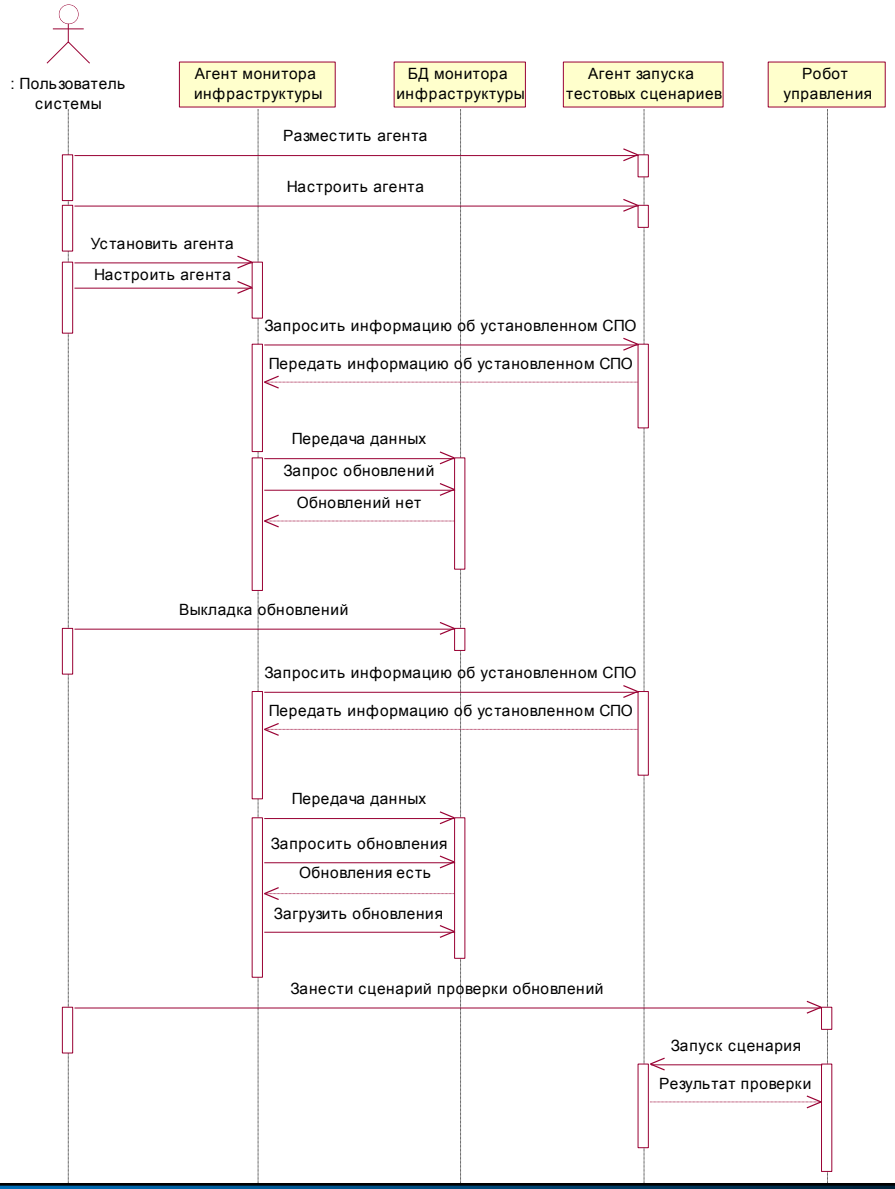
Словарь сущностей

- Agents – Агенты системы
- AgentUpdates – Обновления для агентов монитора инфраструктуры
- Computers – Рабочие станции системы
- Groups – Группы рабочих станций
- Perfomance – Ресурсы оперативной памяти на рабочих станциях
- Products – СПО в системе
- Storages – Ресурсы жестких дисков на рабочих станциях

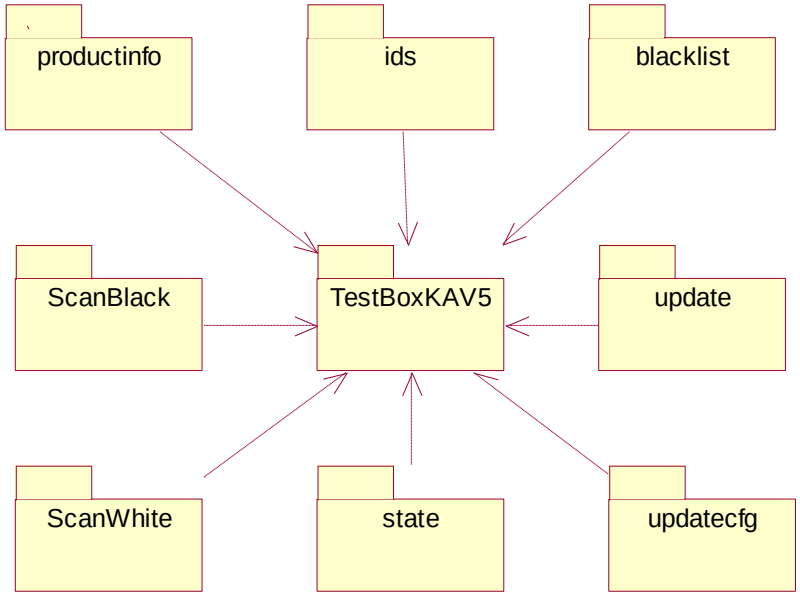
Алгоритм передачи данных



Взаимодействие агентов системы



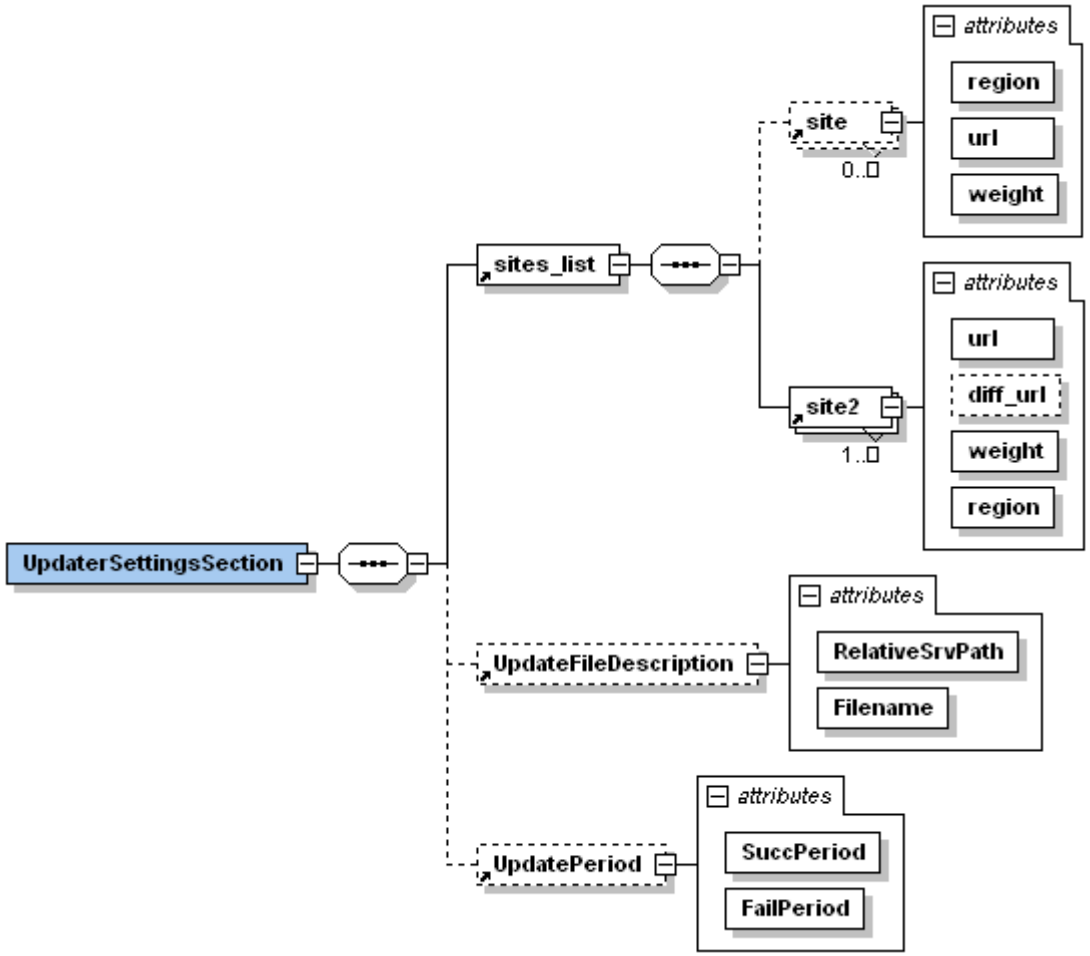
Реализация взаимодействия между пакетами пассивного агента для СПО версии ниже 6.0



Общие коды завершения тестовых сценариев

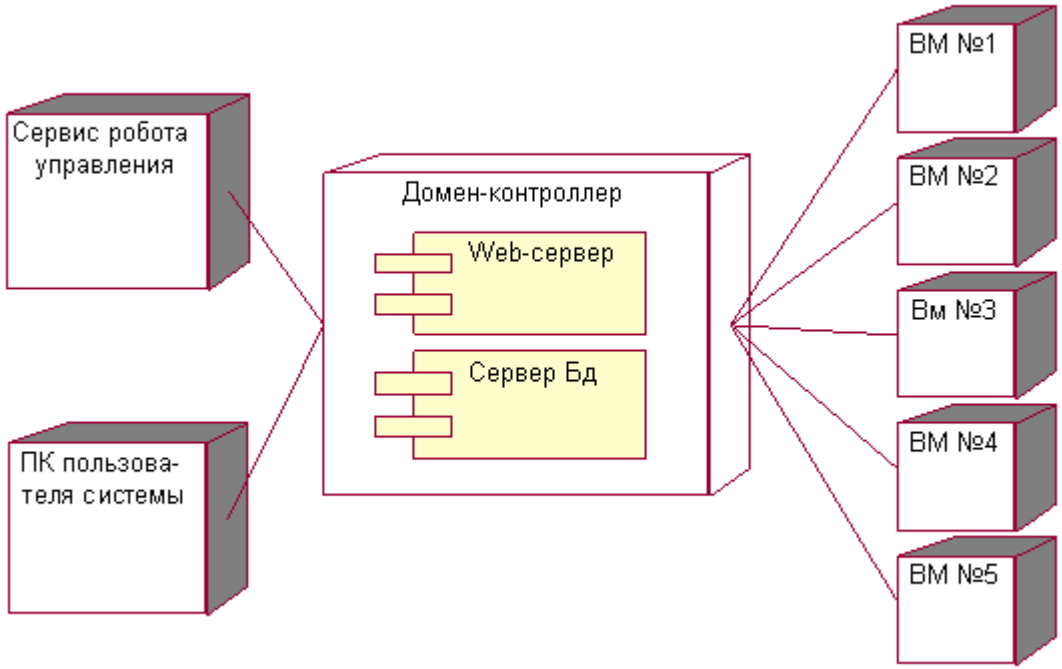
Код завершения	Возвращаемое с кодом значение	Описание
0	TEST_OK	Сценарий успешно завершен
+1	TEST_NEED_RESTART	Требуется перезагрузка рабочей станции
+2	TEST_NOT_PASSED	Сценарий не пройден
+n	TEST_NOT_PASSED	Дополнительные коды возврата для сценариев, проходящих в несколько этапов.
-1	TEST_FAILD	Внутренняя ошибка приложения при выполнении сценария
-2	TEST_INVALID_CMD_LINE	Неверно задан аргумент командной строки
-3	TEST_INVALID_PAR	Неверно задан параметр (несуществующая директория или файл)

Структура файла-схемы для индекса updcfg.xml



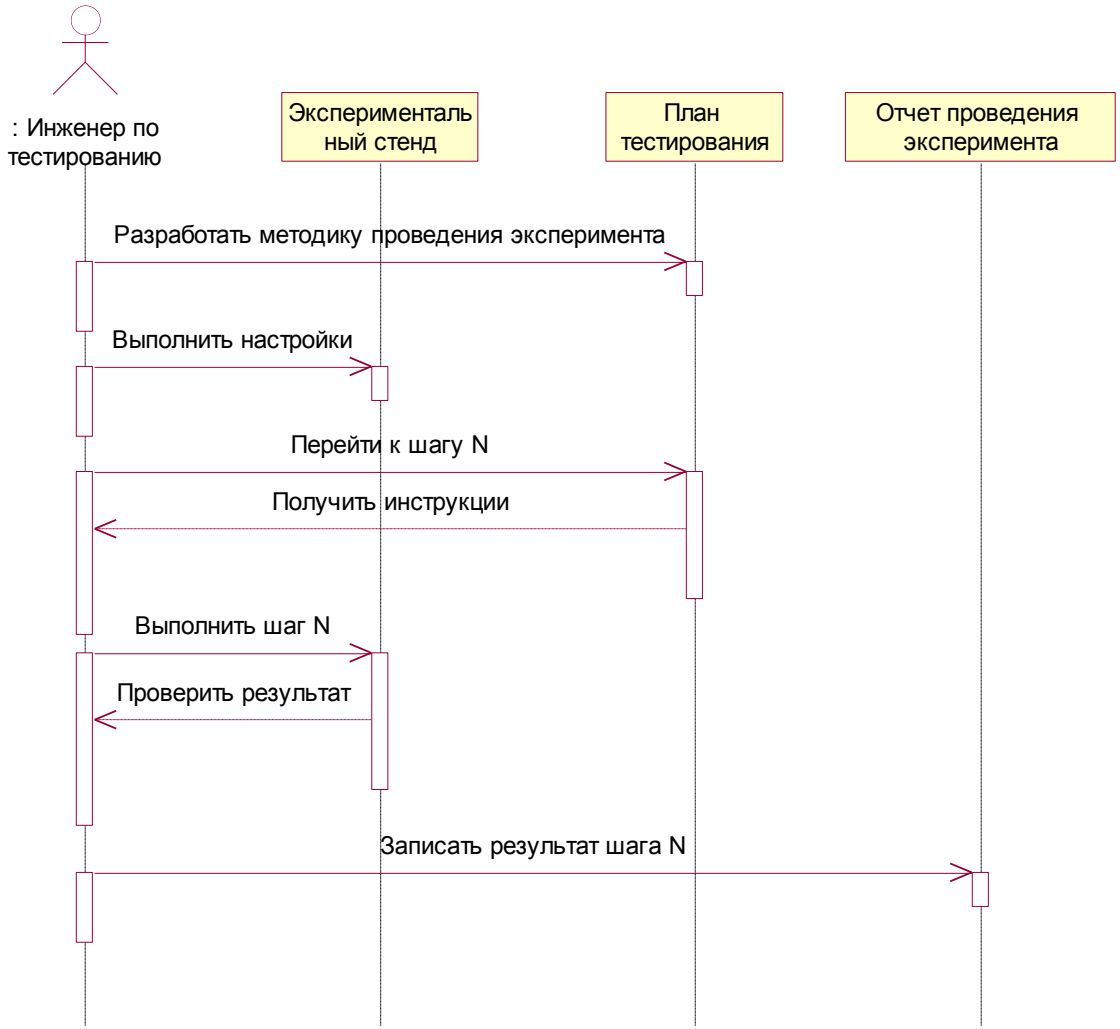
sites_list – обязательный составной атрибут (может быть только один);
UpdateFileDiscription, UpdatePeriod – необязательные сложные атрибуты (может быть только один);
site – необязательный сложный атрибут (может быть несколько);
site2 – обязательный сложный атрибут (может быть несколько, минимальное количество = 1)
region, url, weight, RelativeSrvPath, Filename, SuccPeriod, FailPeriod – обязательные простые атрибуты сложных элементов;
diff_url – необязательный простой атрибут сложного элемента.

Развертывание экспериментального стенда



- Домен-контроллер: Microsoft Windows Server 2003 с ролью контроллера домена, СУБД Microsoft SQL Server 2005 с развернутой БД монитора инфраструктуры, Internet Information Service с сервисами монитора инфраструктуры, ASP AJAX Extention, VMWare Server 1.0.2.
- ВМ №1: Windows XP с установленным KIS6.0.0.303
- ВМ №2: Windows XP с установленным KIS6.0.1.411
- ВМ №3: Windows XP с установленным KIS6.0.2.621
- ВМ №4: Windows Vista с установленным KIS6.0.2.621
- ВМ №5: Windows Vista x64 с установленным KIS6.0.2.621

Последовательность действий при проведении эксперимента



Эксперимент проводился по разработанной методике (плану-тестирования). Результаты эксперимента показали:

- Время проверки обновлений не превышает 30 минут;
- Время восстановления монитора инфраструктуры (web-сервера и сервера БД одновременно) не превышает 30 минут;
- Модули системы работают автономно и выход из строя одного не приводит к неработоспособности других модулей.

Апробация

- Система внедрена в инфраструктуру обновлений ЗАО “Лаборатория Касперского” и функционирует на 40 рабочих станциях.

Выводы

- Разработаны модули системы: монитор инфраструктуры, пассивные и активные агенты, модуль проверки индексов; доработан робот управления инфраструктурой обновлений.

- Разработана концепция построения системы в будущем: модуль сомодиагностики и дополнительные требования к разрабатываемому СПО.

- Развернут экспериментальный стенд и проведены экспериментальные исследования работоспособности системы. Основываясь на результатах экспериментов было принято решение о внедрении системы в инфраструктуру обновлений.