Согласно техническому заданию была разработана система удалённого обслуживания рабочих станций под управлением операционной системы Windows. Система предназначена для удалённой очистки дисков от ненужной информации большого количества компьютеров.

В Microsoft Visual Studio было создано три решения (Solution). Первое решение RemoteTasksServer отвечает за реализацию центрального сервера. Решение включает в себя два проекта. Первый проект CentralServer реализует WCF-сервисы для обмена данными с другими решениями: RemoteTasksClient и RemoteTasksExecutive. Перечень контрактов сервисов с подробными комментариями представлен в приложении 1. Сервисы реализованы с помощью трёх интерфейсов.

1. Интерфейс для передачи информации об удалённых компьютерах IWcfComputers.
2. Интерфейс для передачи информации о задачах для удалённых компьютеров IWcfTasks.
3. Интерфейс для передачи информации о выполнении задач для удалённых компьютеров IWcfTaskResults.

Remote Tasks Server выступает в роли центрального диспетчерского узла, с которым взаимодействуют как рабочее место оператора Remote Tasks Admin, так и удалённые компьютеры, на которых развёрнута служба Remote Tasks Executive.

Сценарий взаимодействия выглядит следующим образом.

На каждом удалённом компьютере служба Remote Tasks Executive периодически вызывает операцию Update у WCF-сервиса IWcfComputer, через которую передаёт на сервер информацию о своём удалённом компьютере. Передаётся имя компьютера, его IP-адрес и другие сведения, которым соответствуют свойствам класса PrxComputer. Операция Update сохраняет данные о компьютере в базе данных в таблице Computers посредством Entity Framework.

Рабочее место администратора Remote Tasks Admin получает от сервера перечень всех удалённых компьютеров путём вызова операции GetList. Remote Tasks Server извлекает список компьютеров из базы данных через Entity Framework и возвращает эти данные через операцию GetList сервиса IWcfComputers. Когда администратор назначает задачу для удалённого компьютера, приложение Remote Tasks Admin с помощью операции Update посылает информацию о задаче на сервер. Сервер сохраняет задачу в базе данных в таблице Tasks посредством Entity Framework.

Сервис удалённого компьютера Remote Tasks Executive периодически получает список задач от Remote Tasks Server и сохраняет его в файле Tasks.xml (процедура GetTasksFromServerAndSaveToFile). Далее процесс считывает список задач (процедура ReadAndSetTasksFromFile) из этого xml-файла, и выполняет задачи, у которых наступил момент выполнения (метод ExecuteAllTasks, класс TaskProcessor).

Приложение RemoteTasksAdmin предназначено для удалённого администрирования группы компьютеров. В этом приложении присутствует интерфейс для работы с удалёнными компьютерами. Оператор может назначать задачи для удалённых компьютеров и редактировать их. Приложение обращается к WCF-сервисам как для получения информации о компьютерах, задачах и результатов задач от сервера, так и для отправления новых и отредактированных задач на сервер.

Приложение 1. Интерфейсы вызовов WCF-сервисов, которые предоставляет Remote Tasks Server

namespace ProxyClasses

{

/// <summary>

/// Набор сервисов для работы с данными по удалённым компьютерам

/// </summary>

[ServiceContract()]

public interface IWcfComputers

{

/// <summary>

/// Возвращает список компьютеров, с которыми работает администратор

/// </summary>

/// <returns></returns>

[OperationContract()]

List<PrxComputer> GetList();

/// <summary>

/// Сохраняет в базе данных данные о компьютере

/// </summary>

/// <param name="prxComputer">Сериализованный объект типа PrxComputer</param>

/// <returns></returns>

[OperationContract()]

int Update(PrxComputer prxComputer);

/// <summary>

/// Возвращает информацию о компьютере по его Id

/// </summary>

/// <param name="id">Id компьютера в базе данных</param>

/// <returns></returns>

[OperationContract()]

PrxComputer GetItem(int id);

/// <summary>

/// Возвращает информацию о компьютере по его Mac-адресу

/// </summary>

/// <param name="macAddress">Мас-адрес компьютера</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

PrxComputer GetItemByMacAddress(string macAddress);

}

/// <summary>

/// Набор сервисов для работы с задачами для удалённых компьютеров

/// </summary>

[ServiceContract()]

public interface IWcfTasks

{

/// <summary>

/// Возвращает список задач для компьютера по его Id

/// </summary>

/// <param name="computerId">Id компьютера в базе данных</param>

/// <returns></returns>

[OperationContract()]

List<PrxTask> GetList(int computerId);

/// <summary>

/// Возвращает список задач для компьютера по Мас-адресу

/// </summary>

/// <param name="macAddress">Мас-адрес сетевой карты компьютера</param>

/// <param name="activeOnly">Истина, если хотим получить только активные задачи, иначе возвращает все задачи</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

List<PrxTask> GetListByMacAddress(string MacAddress, bool activeOnly);

/// <summary>

/// Сохраняет задачу в базе данных сервера

/// </summary>

/// <param name="prxTask">Сериализованный объект, в котором передаётся задача</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

int Update(PrxTask prxTask);

/// <summary>

/// Возвращает задачу по её Id в базе данных сервера

/// </summary>

/// <param name="id">Id задачи в базе данных</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

PrxTask GetItem(int id);

}

/// <summary>

/// Набор сервисов для работы с результатами выполнения задач на удалённых компьютерах

/// </summary>

[ServiceContract()]

public interface IWcfTaskResults

{

/// <summary>

/// Возвращает список результатов выполнения задач для компьютера с определённым Id в базе данных

/// </summary>

/// <param name="id">Id компьютера в базе данных</param>

/// <param name="startDate">Дата начала периода, за который выводится список выполнения задачи</param>

/// <param name="endDate">Дата конца периода, за который выводится список выполнения задачи</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

List<PrxTaskResult> GetListForComputer(int id, DateTime startDate, DateTime endDate);

/// <summary>

/// Возвращает список результатов выполнения для задачи с определённым Id в базе данных

/// </summary>

/// <param name="id">Id задачи в базе данных</param>

/// <param name="startDate">Дата начала периода, за который выводится список выполнения задачи</param>

/// <param name="endDate">Дата конца периода, за который выводится список выполнения задачи</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

List<PrxTaskResult> GetListForTask(int id, DateTime startDate, DateTime endDate);

/// <summary>

/// Возвращает список резутатов выполнения всех задач за определённый период

/// </summary>

/// <param name="startDate">Дата начала периода, за который выводится список выполнения задачи</param>

/// <param name="endDate">Дата конца периода, за который выводится список выполнения задачи</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

List<PrxTaskResult> GetList(DateTime startDate, DateTime endDate);

/// <summary>

/// Сохраняет результат выполнения задачи в базе данных сервера

/// </summary>

/// <param name="prxTaskResult">Информация о результате выполнения задачи</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

int Update(PrxTaskResult prxTaskResult);

/// <summary>

/// Возвращает результат выполнения задачи по его Id в базе данных

/// </summary>

/// <param name="id">Id результата выполнения задачи в базе данных</param>

/// <returns></returns>

[OperationContract()]

[FaultContract(typeof(PrxSimpleException))]

PrxTaskResult GetItem(int id);

}

}