

РОЗРОБКА АГЕНТНОЇ ПЛАТФОРМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ WINDOWS COMMUNICATION FOUNDATION

У даній роботі проводиться аналіз існуючих програмних засобів побудови агентних платформ. Обґрунтовано доцільність вибору WCF як засобу організації комунікації. Доведено підвищення швидкості роботи платформи за даною реалізацією.

***Ключові слова:** агент, агентна платформа, Java RMI, Microsoft Windows Communication Foundation, Microsoft Remoting.*

В данной работе проводится анализ существующих программных средств построения агентных платформ. Обоснованно целесообразность выбора WCF как средства организации коммуникации. Доказано повышение скорости работы платформы за данной реализацией.

***Ключевые слова:** агент, агентная платформа, Java RMI, Microsoft Windows Communication Foundation, Microsoft Remoting.*

This paper deals with the analysis of existing programming development tools of agent platforms. The suitability of the choice of WCF as means for communication organization is grounded. The increase of speed of the platform work with this particular realization is proved.

***Key words:** agent, agents platform, Java RMI, Microsoft Windows Communication Foundation, Microsoft Remoting.*

ВСТУП

В останні роки широкого розповсюдження набули дослідження розподілених застосувань. Особливо це стосується мережних застосувань в Інтернеті, елекомунікаціях, базах даних. Використання при цьому агентного підходу дає такі основні перспективи: суттєве зменшення кількості трафіку, що передається мережею; прискорення процесів збору, обміну і обробки інформації у мережі; зменшення навантаження на серверні апаратні засоби; збільшення надійності; зменшення ймовірності виникнення вузьких місць або так званих BottleNeck; збільшення ресурсних можливостей [1].

Стандартизація використання програмних засобів підтримки агентних платформ є важливою темою у контексті розвитку агентних технологій, який відбувається досить однобічно. В основному тут панують безальтернативні застосування побудовані на Java. Тому бачиться доцільним розгляд інших підходів.

Однією з таких альтернатив є мова програмування C# і платформа .Net. Ці засоби є досить поширені у спільноті розробників Windows сумісних застосувань, але нам не відоме їх використання для створення агентних засобів, не дивлячись на те, що цією технологією надаються зручні ієрархії класів і типи даних для застосувань. Зупинимося на особливостях такого використання в данній роботі.

Основними інструментами побудови агентної платформи є Java RMI, Microsoft Windows Communication Foundation, Microsoft Remoting. Зупинимося на технології .Net Windows Communication Foundation [2]. Зручність використання останньої зумовлений наступними критеріями: інтероперабельність (дає можливість співпрацювати з платформами побудованими на інших операційних системах і засобах комунікації); зручність використання і налаштування необхідних параметрів зв'язку і функціонування; підтримка загальнопоширеної і зручної мови

C# і .NET платформи, а також кодів, описаних іншими мовами програмування, що входять до списку Common Language Runtime; оптимальність використання ресурсів апаратної платформи.

Проаналізуємо детальніше використання зазначеного інструментарію.

1. ФУНКЦІОНАЛЬНІСТЬ .NET REMOTING, ЇЇ НЕДОЛІКИ

.NET Remoting представляє специфічний програмний інтерфейс для організації міжпроцесної комунікації. Ця технологія дозволяє застосуванню створювати об'єкти, що є доступними у різних доменах застосувань, процесах і розташованих на різних комп'ютерах з'єднаних мережею. NET Remoting надає посилання на віддалений об'єкт клієнту, далі клієнтське застосування використовує цей об'єкт як свій локальний об'єкт. Віддалені об'єкти ідентифікуються через URL комп'ютера, який містить цей об'єкт. Клієнтське застосування має знати інтерфейс доступу до цього об'єкта. Всі викликані методи об'єкта серіалізуються у потік байтів і передаються на хост об'єкта через канал, що залежить від використаного протоколу зв'язку. На хості ці методи викликаються як локальні без використання довгих програмних конструкцій. Повернені результати цих методів знову серіалізуються у потік байтів і передаються клієнту.

Одною з переваг .Net Remoting є простота використання з точки зору розробника.

2. ЗАСОБИ, ЩО НАДАЮТЬСЯ JAVA

Технологія Java надає Remote Method Invocation, як інструмент побудови розподілених Java-to-Java застосувань. За допомогою RMI методи класів одного хосту можуть бути викликані з іншого хосту, можливо навіть іншого комп'ютера. RMI використовує серіалізацію для маршалінгу і анмаршалінгу параметрів функцій. Однією з переваг цієї технології є можливість передавати визначення класів у разі їх відсутності в кодї клієнтського застосування. У цьому випадку поведінка і параметри об'єктів залишаються незмінними при їх пересилці на клієнтське застосування.

Однією з позитивних характеристик Java RMI є так званий Http Tunneling. Ця функціональність дозволяє RMI самостійно обирати протокол з'єднання з декількох можливих.

RMI не є ефективним при використанні у системах постійного обміну повідомленнями. Ця технологія призначена для використання при розв'язанні задач типу «виклик – відгук». Тому технологія не є найкращим варіантом для побудови агентної платформ з великим об'ємом обміну повідомленнями.

Проблемним є також неможливість організації комунікації з іншими окрім Java застосуваннями. Співпраця агентів різних агентних платформ у мережі стає неможливою.

До недоліків цієї системи можна додати не стабільну надійність передачі даних. Методи на віддалених застосуваннях є синхронними, тому у разі відмови мережі застосування можуть бути призупинені або «заморожені».

3. WCF ЯК НАСТУПНИЙ КРОК РОЗВИТКУ .NET REMOTING

WCF – програмний каркас розробки застосування з можливістю кооперування. WCF є комунікаційною частиною .Net Framework. Вона була розроблена на базі .NET Remoting. Застосування, що використовують ці технології можуть співпрацювати, тому WCF не вимагає обов'язкового переходу, а лише пропонує нові можливості для розробників [2, 3, 4].

Використання технології WCF забезпечує такі переваги.

Можливість використання WCF сервісів застосуваннями, які побудовані за допомогою інших технологій. Сервіси WCF є описаними у стандартній формі веб-серверних сервісів. Конфігураційні дані сервісу є описаними у файлі метаданих з розширенням .config, який є XML сумісний документ, доступний при виклику MetadataExchange (MEX) частини сервісу. Цей файл має використовуватись будь-якою клієнтською частиною для конфігурування параметрів зв'язку, типу транспорту, адрес і характеристик кінцевих точок (endpoint), прив'язок, поведінок і контрактів сервісу.

Можливість надання сервісів як сервісів Windows, окремого застосування, сервісу сервера IIS. У кожному зі способів хостингу сервісів є свої переваги і можливості надані розробнику. Хостинг у власному застосуванні є простим методом, який не передбачає попереднє налаштування серверних параметрів і конфігурування серверної частини. Але керувати таким

типом хостингу можна лише через код програми, зкомпільований за необхідності зміни якихось параметрів серверу (порти доступу, глобально доступні папки, права доступу до файлів, реакція на специфічні веб запити). IIS надає всі ці можливості, але він є досить важким у налаштуванні і адмініструванні.

Бінарні повідомлення WCF є коротшими за згенеровані Remoting, що збільшує швидкість обміну повідомленнями між клієнтом і сервером. Завдяки оптимізації заголовків повідомлень, якими обмінюються сервер і клієнт, у WCF була значно покращена швидкість обміну даними у порівнянні з .Net Remoting.

Надається можливість розширення сервісів. Сервіс як програма має змінюватись у процесі розробки або реалізації покращеної функціональності. Але водночас різні версії сервісів мають бути підтримувані і працювати через можливість існування клієнтської частини, що залежить від специфічної частини сервісу і яка вже є зовнішньою впровадженою версією програмного продукту.

Серіалізація може бути перевантажена у випадку WCF. Серіалізація об'єктів у форму, придатну для передачі по мережі, може бути перевантажена іншим типом серіалізації використаним розробником. Це може стати у нагоді в наступних випадках: необхідно мати специфічний безпечний формат передачі об'єктів від серверу до клієнта, потрібна оптимізація з точки зору розміру довжини повідомлення, або часу на серіалізацію об'єктів.

WCF краще підтримує сервіс-орієнтовані сценарії. Наразі сервіс-орієнтований підхід є широко впровадженим у мережних застосуваннях. WCF надає типові сценарії Behaviors, Contract Configurations для комунікації застосувань. Зменшується кількість коду, який необхідно розробити.

Надаються декілька можливих варіантів кодування повідомлень. Повідомлення можуть бути закодовані з використанням алгоритмів шифрування для безпечного обміну. Також можливим і задання користувачем іншого типу кодування, який робить повідомлення закритими для інших застосувань або людини.

Для побудови сервісу використовується звичайна семантика Common Language runtime сумісних мов програмування. Контракти сервісу можуть бути задані будь-якою CLR мовою.

4. ПЕРЕВАГИ ВИКОРИСТАННЯ.NET І C#

Найбільш поширеною альтернативою платформі.NET і C# є Java. Основними критеріями для порівняльного аналізу цих технологій вибрані швидкість виконання скомпільованої програми, зручність використання розробником, кількість коду програмування взаємодій. Java Virtual Machine може порівнюватись з .NET CLR, що виконує MSIL код, на відміну від байткоду Java. Java-байткод є інтерпретованим у процесі виконання віртуальною машиною Java. Але завдяки JIT компілятору в .NET застосування виконуються швидше ніж інтерпретований код Java.

.NET Framework підтримується близько двадцятьма мовами програмування на відміну від Java технології, що надає перевагу вибору зручного інструмента розробником. C# включає основні відмінності Java від C та C++ (відсутність вказівників, наслідування від декількох класів, збирання «сміття»-GarbageCollection) [5]. В C# можливе використання некерованого коду. Тому за необхідності використання вказівників і безпосереднього керування життєвим циклом об'єкта розробник може скористатися цією можливістю.

Наявність чи відсутність важливих типових конструкцій даних у обох варіантах зображується у порівняльній таблиці 1:

Таблиця 1

Наявність типів даних у мовах порівняння

Тип даних	Java	C#
Структури	Не підтримується	Репрезентує тип даних після слова struct
Enumeration	Не підтримується	Репрезентує тип даних після слова enum
Делегат	Не підтримується	Represents reference data type with key word delegate(function pointer)

Продовження табл. 1

Клас	Репрезентує шаблон чи вказівниковий тип даних після ключового слова class	Репрезентує вказівниковий тип даних після ключового слова class
Інтерфейс	Репрезентує вказівниковий тип даних після ключового слова interface	Репрезентує тип даних після ключового слова interface
Вказівники	Не підтримується	Підтримується тільки у некерваному коді

ЗАКЛЮЧЕННЯ

Виходячи з зазначених аргументів, можна зробити висновок, що рf критерієм швидкості виконання кінцевого коду та зручності використання мови C# разом із .NET з великою вбудованою бібліотечною базою і широким набором функціональності має претензії на лідерство при виборі інструменту програмування платформ підтримки агентних систем мобільного типу з застосуванням WCF для організації комунікації між агентами.

Оптимальність використання такої платформи забезпечує також зручність створення власного агента для цієї платформи, інтегрованийність з іншими агентними і програмними платформами. Суттєвим фактором є можливість співпрацювати з платформами, побудованими на Java. Ці аргументи мають вплинути на децентралізацію світу агентних платформ [1, 6, 7] і розширення використання інших засобів для їх побудови.

У подальшому бажаним є розвиток засобів WCF для збільшення зручності використання при розробці агентних застосувань. Такий розвиток може включати в себе розробку класів переносу агентних засобів, організації колективного спілкування, управління працюючими агентними кодами, реалізацію сервісів Directory Facilitator [4] і інших необхідних для існування спільноти агентів у мережі.

Щодо розробок самих агентних платформ з використанням даної технології, то реалізація графічного інтерфейсу керування платформою через веб інтерфейс є досить суттєвою задачею.

ЛІТЕРАТУРА

1. Глибовец Н.Н. Использование агентных технологий в системах дистанционного образования // Управляющие системы и машины. – 2002. – № 6. – С. 69-76.
2. «Creating of user-friendly mobile services personalized for tourism», Jani Leinonen, Sasu Tarkoma, Information Society Technologies, 1999.
3. Deploying an Internet Information Services-Hosted WCF Service, Microsoft, www.msdn2.microsoft.com, 2007.
4. «Exploring Windows Communication Foundation».
5. http://www.codeproject.com/KB/WCF/edujini_wcf_scart_01.aspx
6. Comparison of C# and Java: A Developer Perspective, Dare Obasanjo, 2001
7. Глибовец Н.Н. Использование JADE (Java Agent Development Environment) для разработки компьютерных систем поддержки дистанционного обучения агентного типа // Электр. журнал «Образовательные технологии и общество». – 2005. – С. 325-345.
8. «Wandering Code Write Mobile Agents In .NET To Roam And Interact On Your Network» – Matt Neely, www.msdn.microsoft.com/en-us/Magazine, 2006.

Рецензенти: д.т.н., проф. Данілов В.Я.
д.т.н., проф. Фісун М.Т.

© Гороховський С.С., Глибовець А.М.,
Рахматулов Р.Р., 2009

Стаття надійшла до редколегії 16.03.09