Architecture of the Object Oriented ARC framework for C# over .NET

T. Vamsi Kalyan & R.K. Joshi

Department of Computer Science and Engineering
IIT Bombay

Project sponsored by Microsoft

A Presentation for SoDA, Jan 2004, Taj West End
Bangalore
Outline of the talk

• What is ARC

• ARC Kernel Services

• Architectural view of ARC framework

• ARC from the user’s point of view

• Summary and current status of work
Anonymous Remote Computing (ARC)

• ARC is a message passing paradigm to support writing parallel/distributed programs.

• ARC provides parallel/distributed programming constructs.

• Issues in ARC framework are
  ➢ Selecting a processor.
  ➢ Sending a task and waiting for the results.
  ➢ Handling of node failures.
What is ARC over C#/.NET

• Object-oriented, message passing framework to support development of distributed applications in presence of load, heterogeneity, failures and mobility.
• Integration of
  – Anonymity
  – Service orientation
  – Mobility and distribution of objects
• Scalable (dynamic join and dynamic leave) system in terms of participating nodes in a network.
ARC Kernel Level Services

- Registration Service
- Migration Service
- Synchronization
- Retraction Construct
- Connect Construct
- Multi-hopping
- HPF Service
- Fault Tolerance Service
- Auto Execution Service
- Failure Detection
- Object Arrival Intimation Service
- Activation and Deactivation Service
- Join Operation
- Leave Operation
User’s Perspective

• A platform above .NET to support development of distributed applications.

• ARC Kernel level provides partial support for object migration and activating on remote nodes.

• User has to write the methods that are not only application specific but also interact with the framework.

• Some functionalities are provided to the user as partially implemented classes in ARC user lower level.
Design of Container

```
<<interface>> IRefCount
+IncRefCount(): void
+DecRefCount(): void
+RefCount(): int
```

```
<<interface>> IMyObject
+myOperation(param:type): type
```

```
<<interface>> IPushCommand
+push(machine:HPFValue): int
```

```
<<interface>> ISync
+Sync(): void
+GracefulRetract(): void
+connect()
```

```
<<interface>> IMigratable
```

```
<<interface>> IMYObject,
<<interface>> IPushRequest
```

UserProgram

Factory

<<instantiate>> Container

<<use>> «use»

<<use>> «use»

<<class>> Container

<<class>> Factory

<<class>> UserProgram

<<interface>> PContainer

<<implementation>> 2

To Interfaces IMYObject, IPushRequest
Design of Class Real

```
+GetObject(): Object «interface» ITrigge
+Trigger(): void +OnReturn(): void
+OnRetract(): void «interface» IRealUtil
+isRetractionSet(): bool +RegisterForRemoteAccess(): void

+myOperation(param:type): type «interface» IMyObject
+push(machine:HPFValue): int «interface» IPushRequest

+IncRefCount(): void +DecRefCount(): void +RefCount(): int «interface» IRefCount

「interface」 IMigratable
+push(machine:HPFValue): int «interface」 IPushRequest
+Trigger(): void +OnReturn(): void
+OnRetract(): void «interface」 ITrigger

Real
Proxy
FrontEndReal
PReal
IMyObjectMigratable, IPushRequest, ITrigger, IRealUtil

Activator (.NET Remoting)
+GetObject(): Object «instantiates»

Nonoriginator Context
Container at Originator «use»

Microsoft, .NET Framework (version 4.7.0)
```
Component View of ARC Software over LAN
Components Categorized Based on Services

<table>
<thead>
<tr>
<th>ARC Object Service Components</th>
<th>Developer Service Components</th>
<th>Node Administrator Service Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCSSystem</td>
<td>HPFServer</td>
<td>JoinServer</td>
</tr>
<tr>
<td>ObjectServer</td>
<td>FTSServer</td>
<td>Startup Program</td>
</tr>
<tr>
<td>Code Motion Server</td>
<td>Name Server</td>
<td></td>
</tr>
</tbody>
</table>
Summary

• ARC framework integrates
  – Anonymity
  – Mobility and distribution of objects
  – Service orientation at interface and interaction level
  – Object orientation at design and implementation level
  – OOAD process followed for development

• LAN and Internet versions of ARC software code
  – Tested using four Windows 2000 workstations.

• Main design goal was separation of system’s concerns from the programmer’s concerns while building service oriented and object-oriented ARC framework over C#/.NET.
Current Status of the Research

• ARC over LAN has been extended to work over the Internet by keeping reuse and providing same abstraction to the user as main objectives.
  – ASP.NET, XML Web services, .NET, C#

• Security issues are not addressed in current work and needs to be addressed.

• Application framework and application that involve real world problems
Website:

http://www.cse.iitb.ac.in/~rkj/arc....

Thank You