Generation of Distributed System Test-beds from High-level Software Architecture Descriptions

John Grundy, University of Auckland, NZ
Yuhong Cai, University of Auckland, NZ
Anna Liu, CSIRO, Sydney, Australia

ASE 2001

Outline

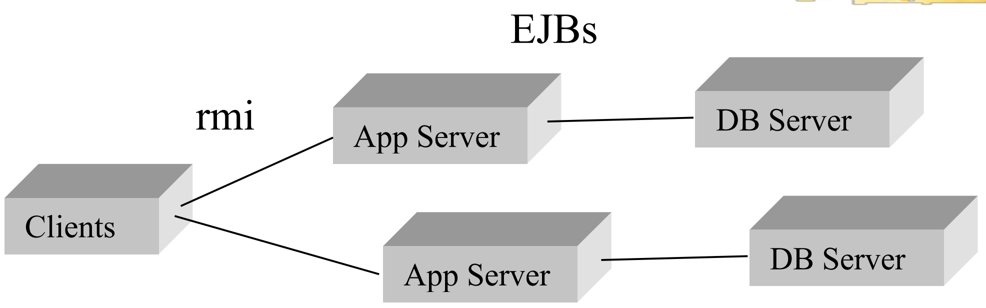
- Motivation
- Our Approach
- Modelling architectures with SoftArch/MTE
- Generating test-bed code, deployment files etc
- Visualising performance results
- Experiences with SoftArch/MTE
- Future work directions

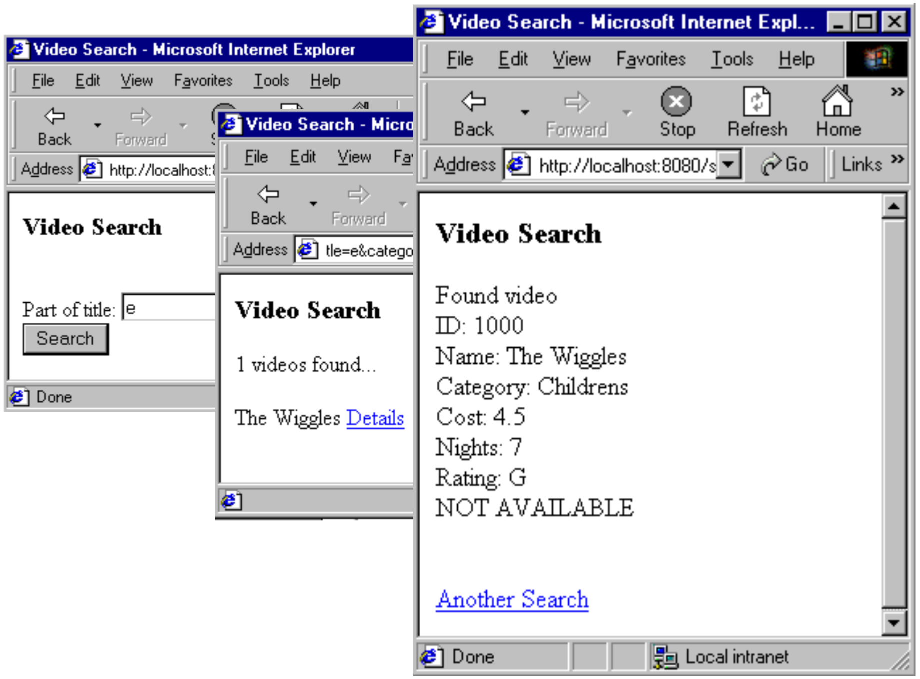
Motivation

- ❑ Distributed system performance evaluation
- ❑ Complexity: new software architectures; middleware; database management; UI technology
- ❑ How do we evaluate DS performance???
  - Rapid prototyping
  - Software architecture-based simulation
  - Existing system evaluation

- ❑ Bottom line: its HARD/time-consuming…

Example...




Video Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward
Address http://localhost:
Done
Video Search - Micro
File Edit View Fa
Back Forward
Address tle=e&catego
Video Search - Microsoft Internet Expl...
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home
Address http://localhost:8080/s
Go
Links
Done
Local intranet

Video Search

Part of title: e

Search

Video Search

1 videos found...

The Wiggles Details

Video Search

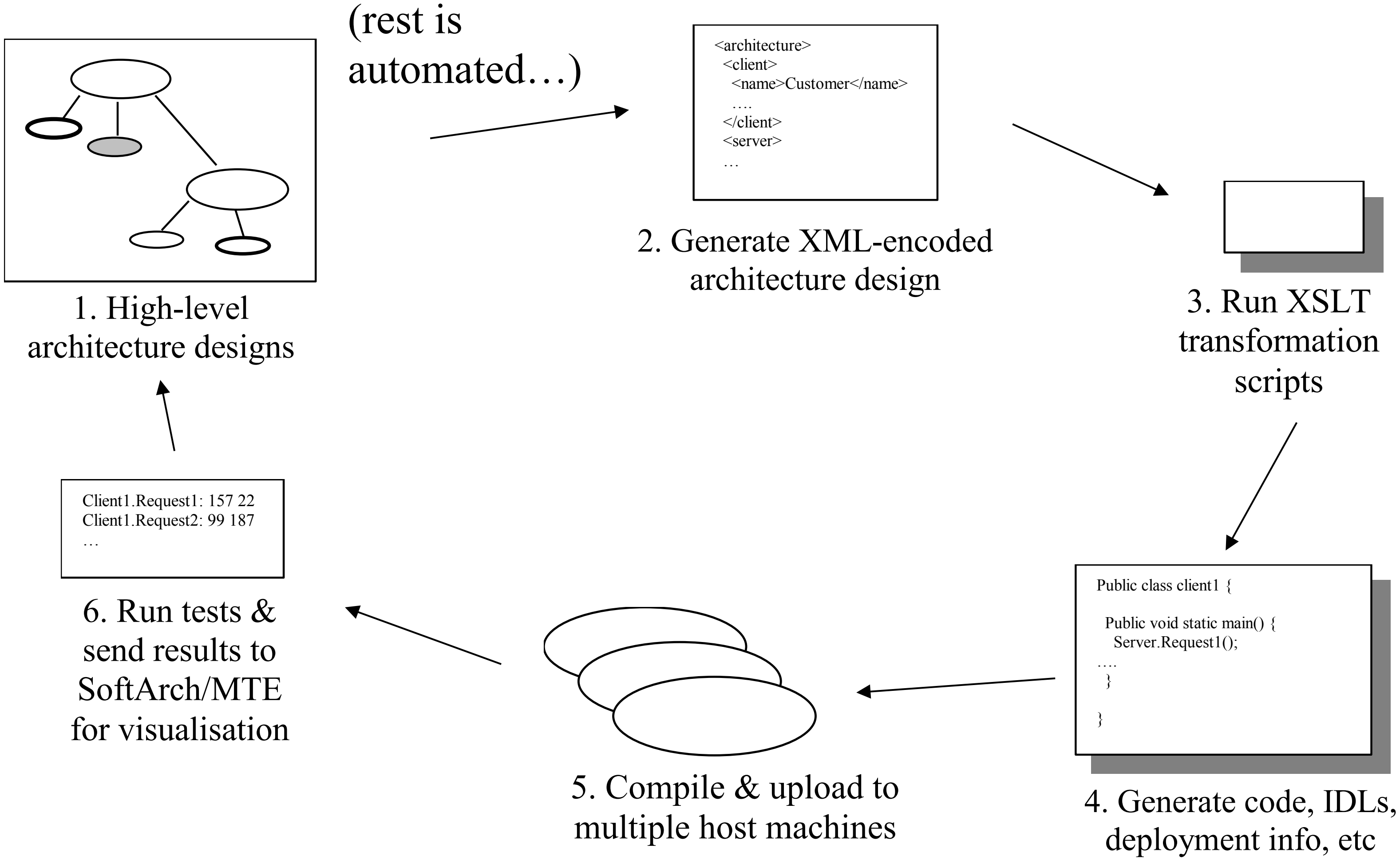
Found video
ID: 1000
Name: The Wiggles
Category: Childrens
Cost: 4.5
Nights: 7
Rating: G
NOT AVAILABLE

Another Search

- ❑ On-line video system
- ❑ Search for videos
- ❑ Rent/return videos
- ❑ Maintain data

- ❑ Choices:
  - Architectures
  - Middleware/DBs/UIs

Our Approach...



(rest is automated…)

1. High-level architecture designs

```
<architecture>
  <client>
    <name>Customer</name>
    ….
  </client>
  <server>
  …
```

2. Generate XML-encoded architecture design

3. Run XSLT transformation scripts

```
Public class client1 {

  Public void static main() {
    Server.Request1();
  ….
  }

}
```

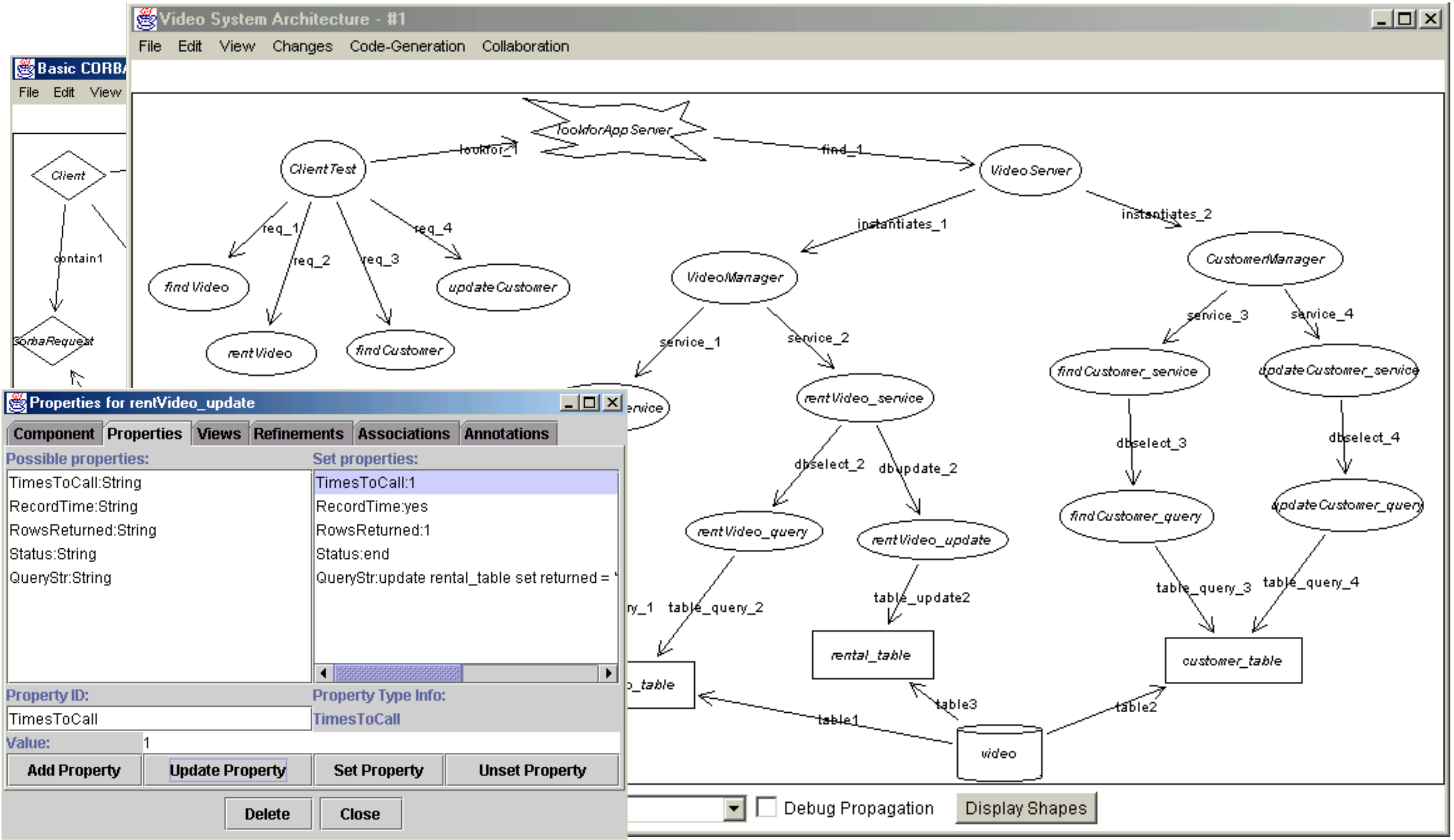
4. Generate code, IDLs, deployment info, etc

5. Compile & upload to multiple host machines

```
Client1.Request1: 157 22
Client1.Request2: 99 187
…
```

6. Run tests & send results to SoftArch/MTE for visualisation

Modelling Architectures in SoftArch/MTE



Possible properties:

TimesToCall:String
RecordTime:String
RowsReturned:String
Status:String
QueryStr:String

Set properties:

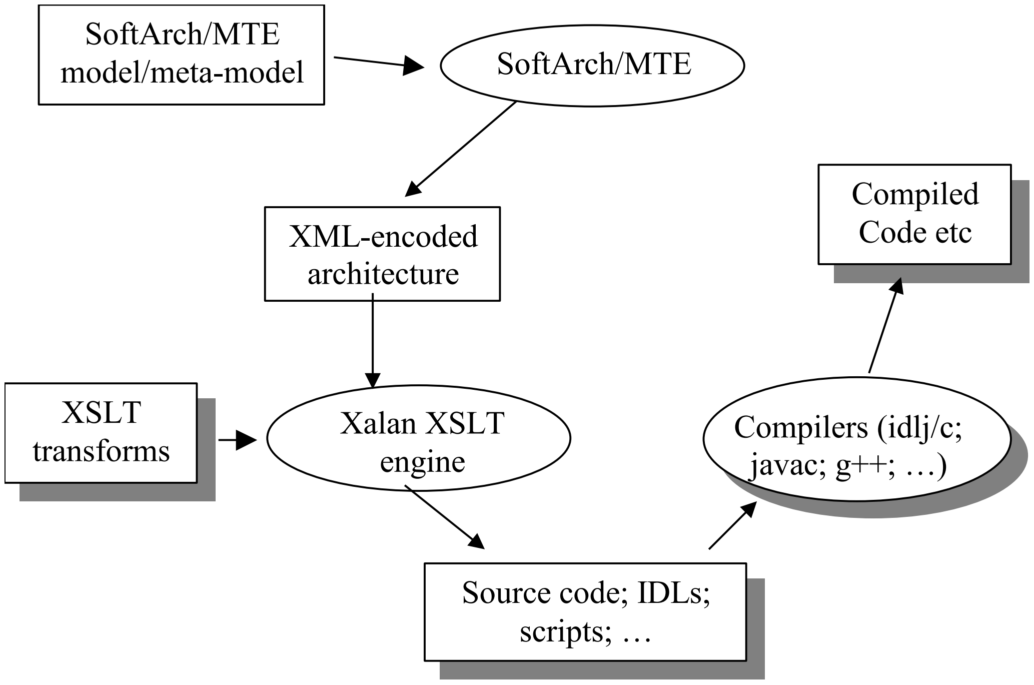
TimesToCall:1
RecordTime:yes
RowsReturned:1
Status:end
QueryStr:update rental_table set returned = '

Property ID: TimesToCall

Property Type Info: TimesToCall

Value: 1

Code Generation


SoftArch/MTE model/meta-model
SoftArch/MTE
XML-encoded architecture
XSLT transforms
Xalan XSLT engine
Source code; IDLs; scripts; …
Compilers (idlj/c; javac; g++; …)
Compiled Code etc

- Model architecture using SoftArch/MTE visual language
- XML encoding of architecture generated
- XSLT scripts used to generate .java, .bat, .xml etc files (see egs in paper…)

XSLT Example

Client_Clie

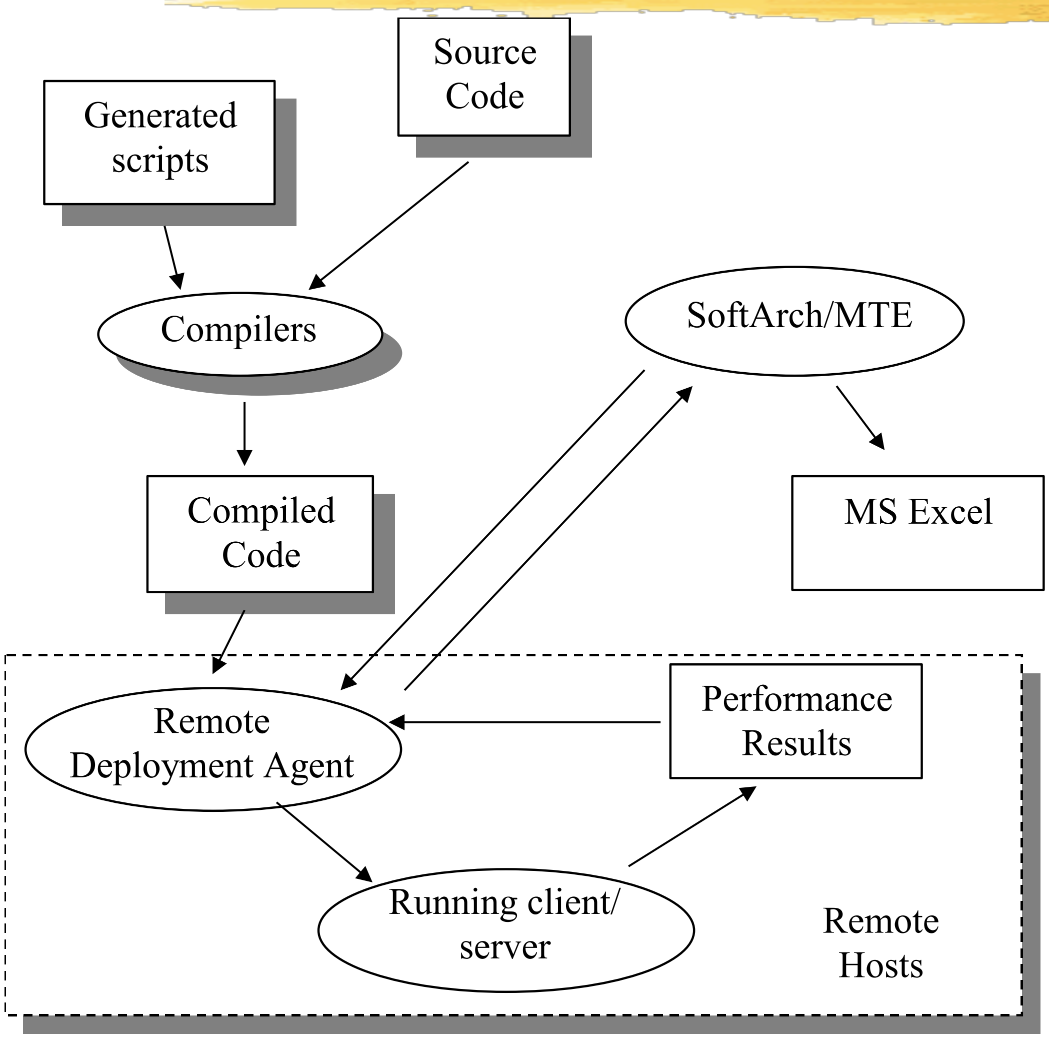
```
<?xml version="1.0" en
<Client>
<Name>ClientTest</Name
<Hosts>LocalHost</Host
<Threads>1</Threads>
<Request>
<Type>CorbaRequest</Ty
<Name>findVideo</Name>
<RemoteObject>VideoMan
<TimesToCall>10</Times
<RecordTime>yes</Recor
</Request>
<Request>
<Name>rentVideo</Name>
<Type>CorbaRequest</Ty
<RemoteObject>VideoMan
<TimesToCall>4</TimesT
<RecordTime>yes</Recor
</Request>
<Request>
…
</Client>
```

```
…
<!-- CORBA reque
<xsl:template ma
    public stati
(<xsl:value-of s
        int iter
  String name =
        String r
        System.g
        long sta
        int i=0;
        while(i
             ser
            i++;
        }
        if(recor
            long
            doub
            Stri
            Syst
            Syst
        }
  }
</xsl:template>

…
```

ClientTest.java

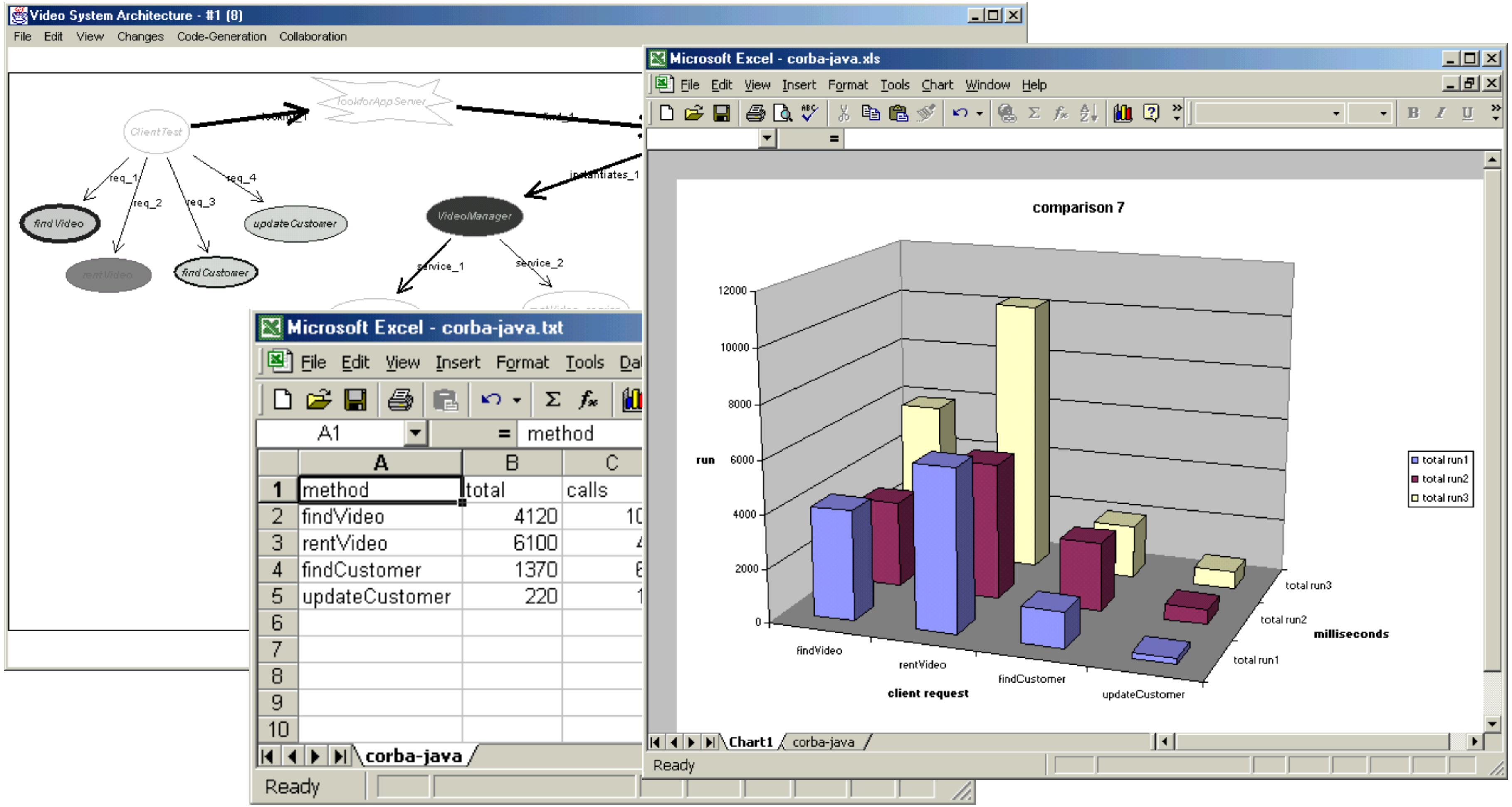
```
public class ClientTest
{
…

   public static void findVideo(VideoManager server){
        int iter = 10;
  String name = "findVideo";
        String recordTime = "yes";
        System.gc();
        long start = System.currentTimeMillis();
        int i=0;
        while(i != iter){
             server.findVideo_service ();
            i++;
        }
        if(recordTime.equals("yes")){
            long time = System.currentTimeMillis() - start;
            double elapse = (double)(time) / (double)
(Math.max(1,iter));
            String perf = name+"\t"+time+"\t"+iter
+"\t"+elapse;
            System.out.println(perf);
            System.err.println(perf);
        }
  }
…
}
```

Running Performance Tests and Viewing Results



- Upload code to hosts (clients and servers)
- Deploy (EJBs)
- Start (RMI, CORBA)
- Run clients
- Results sent back to SoftArch/MTE
- Results displayed

Viewing Performance Results...



	A	B	C
1	method	total	calls
2	findVideo	4120	10
3	rentVideo	6100	4
4	findCustomer	1370	6
5	updateCustomer	220	1
6			
7			
8			
9			
10			

Other Architecture Examples...

- <<GET FROM Yuhong...>>

Experiences...

- Generates 2-, 3-, 4-tier architectures
- Analysed several systems e.g. video, travel system, workflow system
- Very efficient way of obtaining useful performance measures of software architecture+middleware performance
- Figuring out reasons for performance can be hard!
- Hampered by ability of designers to accurately “guess” likely request mixes etc; availability of hosts to run on (threading OK but skews results)
- Need to use throughout development of systems

Conclusions/Future Work

- ❑ SoftArch/MTE demonstrates one-button generate/ compile/deploy/run/ capture/visualise complex architecture performance feasible
- ❑ Provides effective, efficient automated performance testing suite

- ❑ P2P architectures; HTTP/WAP, .NET middleware
- ❑ Incorporate SoftArch/MTE usage in design process
- ❑ Improvements to visualisation of performance
- ❑ Solving performance problems still hard! ☺

References

- Grundy, J.C. Software Architecture Modelling, Analysis and Implementation with SoftArch, In Proceedings of the 34th Hawaii International Conference on System Sciences (Software Architecture Mini-track), Jan 3-6, 2001, Maui, Hawaii, IEEE CS Press, pp. 3826 –3834.
- Grundy, J.C., Mugridge, W.B. and Hosking, J.G. Constructing component-based software engineering environments: issues and experiences, Information and Software Technology Vol 42, No. 2, Special Issue on Constructing Software Engineering Tools, Elsevier Science Publishers.
- Grundy, J.C. and Hosking, J.G. Directions in modelling large-scale software architectures, In Proceedings of the 2000 Conference on Software - Methods & Tools, Wollongong, Australia, Nov 6-10, 2000, IEEE CS Press.
- Grundy, J.C. and Hosking, J.G. High-level Static and Dynamic Visualisation of Software Architectures, In proceedings of the 2000 IEEE Symposium on Visual Languages, Seattle, Washington, Sept. 14-18 2000.
- Grundy, J.C. Construction of an Integrated and Extensible Software Architecture Modelling Environment, In Proceedings of the 2nd International Symposium on Constructing Software Engineering Tools (CoSET'2000), Limerick, Ireland, pp. 51-61.
- Grundy, J.C. Distributed Component Engineering using a Decentralised, Internet-based Environment, In Proceedings of the 3rd ICSE Workshop on Software Engineeering over the Internet, ICSE 2000 Workshop, Limerick, Ireland, June 6 2000, pp. 20-29.
- Grundy, J.C. and Hosking, J.G. Directions in Modelling Large-scale software architectures, In Proceedings of the 1999 Australasian Workshop on Software Architecture, Sydney, Australia Nov 1999.
- Grundy, J.C., Mugridge, W.B. and Hosking, J.G. Visual specification of multiple view visual environments, In Proceedings of IEEE VL'98, Halifax, Nova Scotia, Canada, Sept 1-4, 1998, IEEE CS Press, pp. 236-243.