Lecture 4 ACME Study

- □ Introduction
- □ Arch Design Elements
- □ Attributes
- □ Design Constraints
- □ Type，Pattern and Style

ACME - an ADL developed by CMU

Acme started in 1995 as an ADL interchange language but has evolved to an ADL itself

Three fundamental capabilities:

1) Architectural interchange
2) Extensible foundation for new design and analysis tools
3) Architecture description

Acme language definition and tool developer library (java and C++) available for download. AcmeStudio IDE and HTML documentation generator also available

架构建模语言ACME

ACME － an ADL developed by CMU

ACME is now a typical ADL that also provides model conversion

ACME describes the software architecture in the four views: Structure, Attributes, Constraints, Style

Model Elements:

Seven arch elements: Component, Connector, Port, Role, System, Representation, Re-map

ACME Architectural Elements

Component

the computing and data storage units in a system, such as objects, processes/threads, database servers, etc.

Each component has a group of Port(interface) , with each port defining an interaction point with external environment.

Connectors

An abstraction of the connection or interaction between the components, such as pipes、procedure call、event broadcast, and more complex forms as client/server communication protocol and SQL connector for DB.

Connectors also has the ports that defining the roles that are executed by the corresponding parities involved in the connection.

架构建模语言ACME

Systems

a diagram composed of components and connectors. A system can be a subsystem in a layered architecture while the component and connector can be an independent unit in a system.

Properties

the attribute that describes information on a system and its components or connectors. For instance, the throughput or response delay of a system.

Constraint

the conditions that shall be maintained in various versions of architectural design, which describe the constraints and boundaries of system behaviors. For example, the #connections to a server.

Style

a family of systems with common architectural characters.

架构建模语言ACME

ACME seven architectural model elements

- Component
- Connector
- System
- Port
- Role
- Representations
- Rep-map

架构建模语言ACME

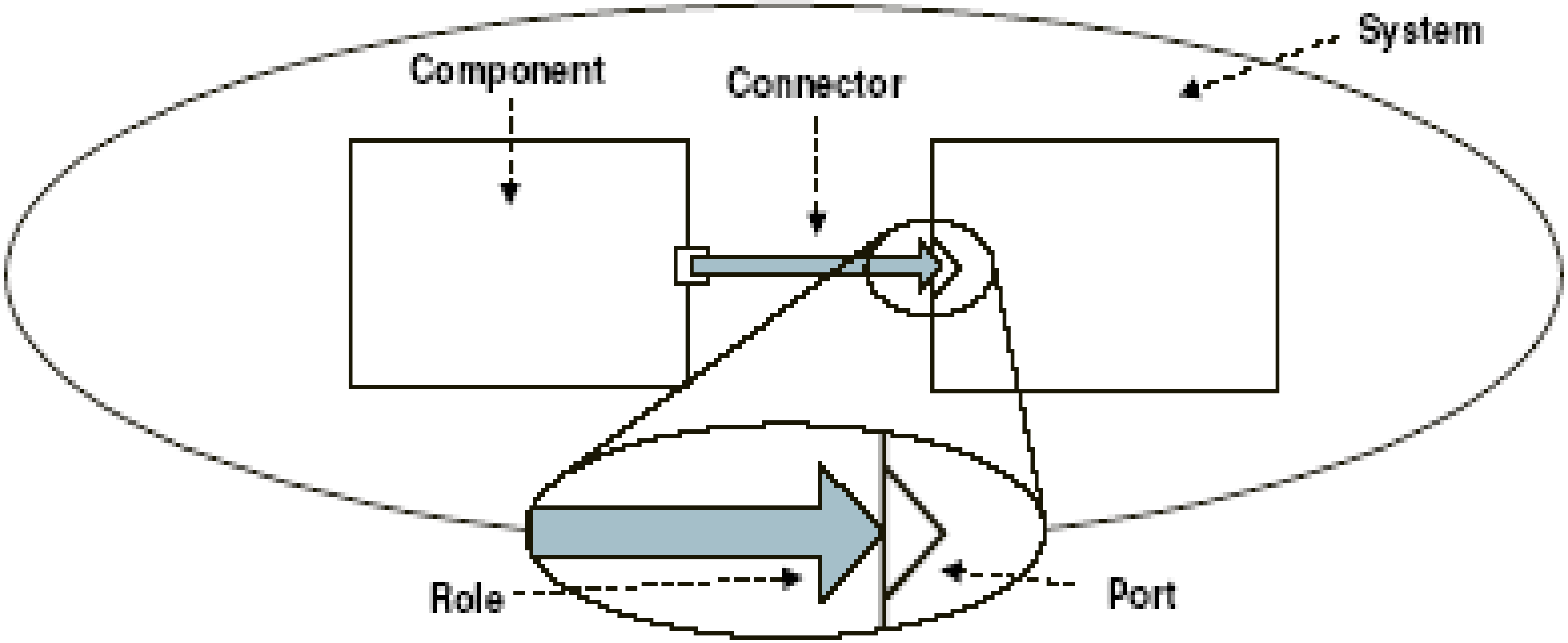
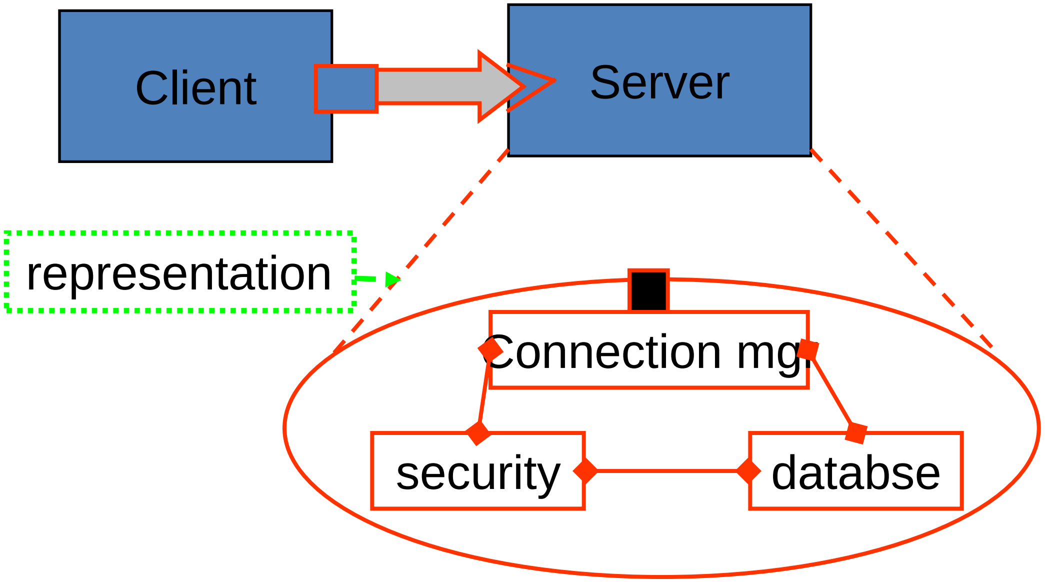


图 2.2: Acme基本元素（引自[GMW00]）

Representation and Rep-map

Representation provides the capability of describing SA at various layers

Rep-map defines the association of the inner structure of represented unit to the external port



C/S Arch Model with representations

Representation

(1) If the representation of a subsystem used, the representation can be considered as the realization of represented subsystem’ port

(2) If the representation used as a diagram, it can be used to replace other diagrams with a simpler presentation。

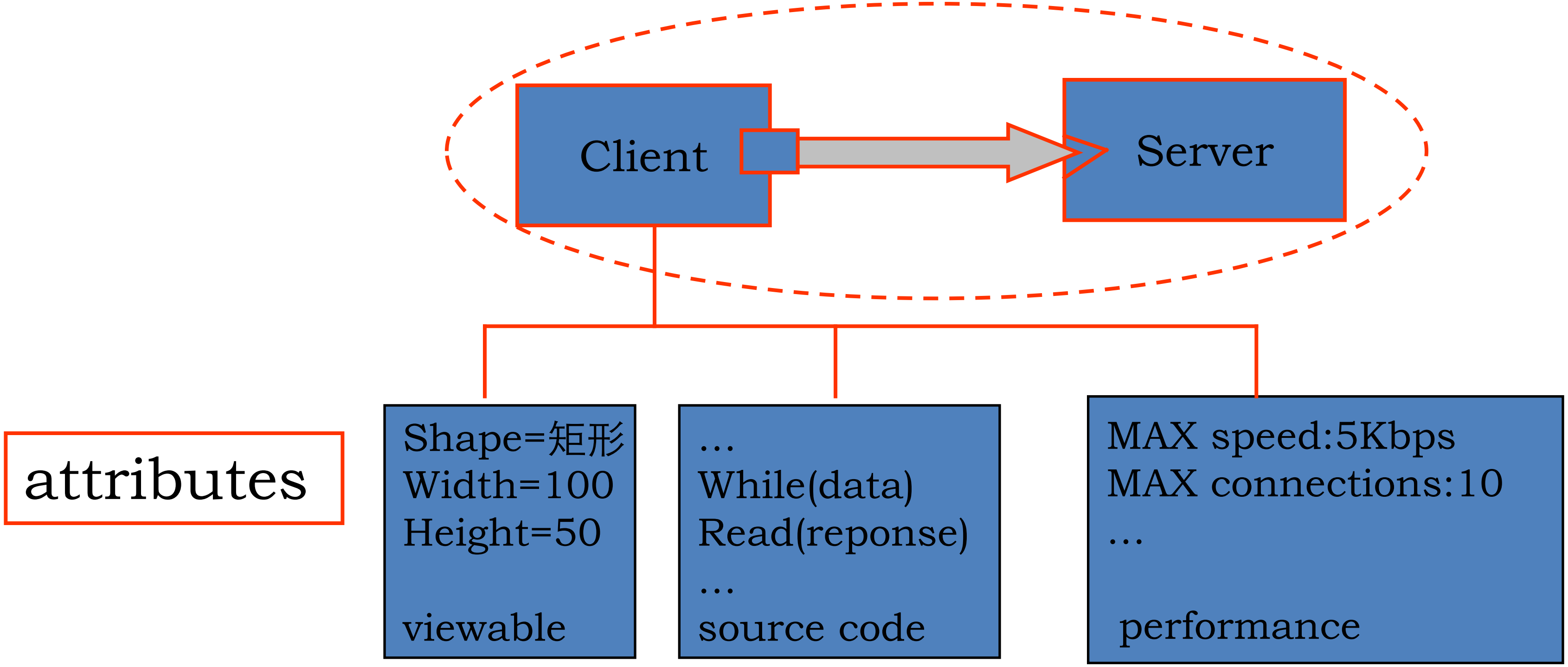
(3) Can also be used to represent a text message or to address the logic boundary for the system

ACME Sample – Description of C/S Architecture

```
System simple_CS = {
    Component client={ Port  sendRequest }
    Component Server={Port  receiveRequest }
    Connector   rpc={Roles{caller,callee} }
    Attachments {
        client.sendRequest to rpc.caller
        server.receiverRequest to rpc.callee
    }
}
```

架构建模语言ACME

ACME Attribute



Client

Server

attributes

Shape=矩形
Width=100
Height=50

viewable

…
While(data)
Read(reponse)
…
source code

MAX speed:5Kbps
MAX connections:10
…

performance

- ACME allows to use an attribute list to describe SA
- An attribute may have name, selective type and value

ACME Sample: scripts with attributes

```
System simple_CS = {
    Component client={ Port  sendRequest
         Properties {requestRate:float=17.0;
                          sourceCode:externalFile=“client.c”}  }
    Component Server={ Port  receiveRequest;
         Properties {macConcurrentClients:integer=1;
                          multithreaded:Boolean=false;
                          sourceCode:externalFile=“server.c }  }
    Connector rpc={Roles{caller,callee}
         Properties{synchronous:boolean=true;
                          maxRoles:integer=2;} }
    Attachments {
         client.sendRequest to rpc.caller
         server.receiverRequest to rpc.callee  } }
```

Design Constraints

- Regulates how the SA design varies and performs
- A constraint may be related to any arch element in the design in a pre-defined range

Samples:

Connected(c1,c2)--- If a connector exists between c1 and c2, it’s true; otherwise it’s false

Reachable(c1,c2)--- If c2 is on the connection path from c1, it’s true; otherwise it’s false

HasProperty(elt,propName)--- If the element elt has an attribute, then name it as propName

Type, Template and Style

(1) Type

- Attribute Type： the type of attributes
- Architectural Type： the style of architecture

```
Component Type Client={        //define component type
      Port Request={Property protocol:CSPprotocolT};
      Property request-rate:Float;
      Invariant size(self.Ports)<=5;
      Invariant request-rate>=0;
      Heuristic request-rate<100;
     }
```

架构建模语言ACME

(2) Template

```
Style client_server = {
    Component Template client(rpc_call_ports:Ports)={
         Port  rpc_call_ports
         Properties {requestRate:float=17.0;
                          sourceCode:externalFile=“client.c”}  }
    Component Template Server(rpc_receive_ports:Ports)={
         Port rpc_receive_ports;
         Properties {macConcurrentClients:integer=1;
                          multithreaded:Boolean=false;
                          sourceCode:externalFile=“server.c }  }
```

架构建模语言ACME

(2) Template (cont'd)

```
Connector Template  rpc(caller_port,callee_port:Port)
     defining(conn:Connector)={ conn=Connector {
          Roles{caller,callee}
          Properties{synchronous:boolean=true;
                             maxRoles:integer=2;} }
          Attachments {
             conn.caller to callee_port
             conn.callee to caller_port  }
          }
}
```

架构建模语言ACME

(2) Template (cont’d)

```
Samples:
System complec_cs:client_server={
          c1=client(send_request);
          c2=client(send_request);
          c3=client(send_request);
          s1=sever(receive_request);
          s2=sever(receive_request);
          rpc(c1.send_request,s1.receive_request);
          rpc(c2.send_request,s2.receive_request);
          rpc(c3.send_request,s2.receive_request);
}
```

架构建模语言ACME

(3) Style

A group of associated templates that form a particular type of SA can be defined as the style of SA, which captures the core set of common features of certain type of SA in the design

Style can also be represented by a Family， which can symbolize a group of systems with same style

架构建模语言ACME

Summary on ACME

- Seven key elements are used to model the SA
- Provide a flexible description tool through representation and rep-cap, which allows the SA model to use external non-formal languages
- Provide the scheme of Type, Template and Family to allow the SA with same style can be modeled and the design can be reused
- Provide an open description framework that contains the primary architectural elements and associated attributes that may be used for measurement, evaluation and improvement

End of Lecture

谢谢！