Internet-based Collaborative System Design Using MOSCITO

André Schneider
Fraunhofer Institute for Integrated Circuits
Design Automation Department

Eero Ivask
Tallinn Technical University

Workshop on Challenges in Collaborative Engineering
Poznan (Poland), April 15-16, 2003
Outline

MOSCITO –
Basic Idea, Applications, Advantages

Fundamental Concept and
Infrastructure

Security Aspects –
Passing Firewalls and Data Encryption
Outline

MOSCITO –
Basic Idea, Applications, Advantages

Fundamental Concept and
Infrastructure

Security Aspects –
Passing Firewalls and Data Encryption
Basic Idea of MOSCITO

Today most computers are connected via networks. Remote resources are available using various techniques ...

1983
rlogin, rsh, rcp, rexec
(Berkeley, UCB)

1989
X11
(Palo Alto, Xerox)
Basic Idea of MOSCITO

Many conventional programs (stand-alone tools, native programs, legacy code) are installed on each computer.

Saber, Matlab, Opal, Ansys, Dymola, AdvancedMS, Spice, Modos, Octave, Scilab, Asterix, Marabu, TSMG, SimulationX, Eldo, TurboTester, EdifImport, VHDL2C, Camad, ...
Basic Idea of MOSCITO

1st step
Collaboration between the tools

(implemented in MOSCITO 2.8)

2nd step
Collaboration between the users

(will be implemented in MOSCITO 3.x)
Basic Idea of MOSCITO

The goal is that a user can combine all needed remote tools (agents) to an appropriated workflow.
Basic Idea of MOSCITO

The goal is that a user can combine all needed remote tools (agents) to an appropriated workflow.
MOSCITO Application 1: Desktop for System Designers

- MOSCITO software must be installed locally.

- A user has access to all available agents in a network.

- The user provides all input and configuration data. (netlists, config files, configuration parameters)

- All agents can be observed by the users front-end. Results are redirected to the front-end.
MOSCITO Application 1: Desktop for System Designers

- MOSCITO software must be installed locally.
- A user has access to all available agents in a network.
- The user provides all input and configuration data. (netlists, config files, configuration parameters)
- All agents can be observed by the users front-end. Results are redirected to the front-end.
MOSCITO Application 2: Applets for e-Learning
MOSCITO Application 2: Applets for e-Learning

A learner can interact with remote simulators via an e-learning portal.
MOSCITO Application 2: Applets for e-Learning

- A user has access to online simulations via a regular, Java-enabled browser.

- A special GUI can be designed and adopted for each application example.

- The applets are useful inside Web-based trainings (WBTs), for online-presentations or for collaboration between project partners.
Advantages of MOSCITO

- design tools can be used remotely
- platform independent
- efficient usage of remote hardware resources
- design tools can be coupled to workflows
- application-specific user interfaces
- secure data transfer
- collaborative work will be supported
Outline

MOSCITO –
Basic Idea, Applications, Advantages

Fundamental Concept and Infrastructure

Security Aspects –
Passing Firewalls and Data Encryption
Main Tasks of MOSCITO Infrastructure

How can a service (an agent) be described?
MOSCITO Agent Descriptor (XML file)

How can a service (an agent) be found in the network?
MOSCITO Master Server (global registry)

How can a service (an agent) be started?
MOSCITO Slave Server (local agent invocation service)
Java object serialization is used for data transfer
Main Tasks of MOSCITO Infrastructure

**Provider**

**How can a service (an agent) be described?**
MOSCITO Agent Descriptor (XML file)

**User**

**How can a service (an agent) be found in the network?**
MOSCITO Master Server (global registry)

**Protocol between provider and user**

**How can a service (an agent) be started?**
MOSCITO Slave Server (local agent invocation service)
Java object serialization is used for data transfer
MOSCITO and Web Services Standards

How can a service (an agent) be described?
**WSDL**
(Web Services Description Language)

How can a service (an agent) be found in the network?
**UDDI**
(Untiversal Deployment and Discovery Interface)

How can a service (an agent) be started?
**SOAP**
(Simple Object Access Protocol)
Infrastructure: MOSCITO Master Server

MOSCITO Master Server maintains a global registry for all agents (services) and users.

Example:
moscito.eas.iis.fhg.de:11001
MOSCITO Master Server maintains a global registry for all agents (services) and users.

An agent can be dynamically added to and removed from registry.

Example:
moscito.eas.iis.fhg.de:11001
MOSCITO Master Server maintains a global registry for all agents (services) and users.

An agent can be dynamically added to and removed from registry.

Example:
moscito.eas.iis.fhg.de:11001
After starting a client program (MOSCITO Desktop) a user will be registered at MOSCITO Master Server.
Infrastructure: MOSCITO Master Server

After starting a client program (MOSCITO Desktop) a user will be registered at MOSCITO Master Server.
Infrastructure: MOSCITO Master Server

After starting a client program (MOSCITO Desktop) a user will be registered at MOSCITO Master Server.

Each user can see the agents and users being online.
Configuration of Workflows and Agents

Now the MOSCITO Desktop helps a user to select an appropriated workflow and to configure the needed agents.
Start der Agenten und Initialisierung des Workflows

The MOSCITO Master Server receives the request for instantiating the workflow and forwards them to the MOSCITO Slave Servers.
MOSCITO Slave Servers start the agents and all connections between the agents and the MOSCITO Desktop can be established.
Start der Agenten und Initialisierung des Workflows

MOSCITO Slave Servers start the agents and all connections between the agents and the MOSCITO Desktop can be established.
Data Transfer

MOSCITO infrastructure allows transfer of any data types (ASCII files, binary files, strings, numbers, objects).

Data items are sent as serialized objects within container objects: MOSCITO Datagrams.
Data Transfer with Container Objects: MOSCITO Datagrams

```java
datagram = new MoscitoDatagram();
datagram.putData("Netlist", netlist);
datagram.putData("x0", new Double(12.5));

socket.write(datagram);

datagram = socket.read();

netlist = (MoscitoFile) datagram.getData("Netlist");
x0 = (Double) datagram.getData("x0");
```
Data Transfer with Container Objects
Data Transfer with Container Objects

An agent gets the needed data from the container (datagram) and puts back the result data to the container.

Data items can be identified by tags (names).
Tool Encapsulation

A Java wrapper (MOSCITO agent) is needed for each integrated tool.

The agent provides all interfaces for data transfer and controlling the tool.
Tool Encapsulation

MoscitoAgent

Each MOSCITO agent works data-driven.

MOSCITO agent receives input datagram, unpacks the needed content and passes them to the encapsulated tool.

After processing data the agent performs the needed conversions and puts the data in the output datagram.
Outline

✓ MOSCITO –
  Basic Idea, Applications, Advantages

✓ Fundamental Concept and
  Infrastructure

→ Security Aspects –
  Passing Firewalls and Data Encryption
Problems: Firewalls and Security

Transfer of sensitive design data via Internet
Solutions for the Firewall Problem

- open all ports
Solutions for the Firewall Problem

- open all ports
- open one port for proxy

Firewall
Proxy
proxy-host:5001
Solutions for the Firewall Problem

- open all ports
- open one port for proxy
- HTTP tunneling via Web server (port 80)
Solutions for the Firewall Problem

- open all ports
- open one port for proxy
- HTTP tunneling via Web server (port 80)
- VPN tunneling (Virtual Private Network)
Solutions for the Firewall Problem

• open all ports
• open one port for proxy
• HTTP tunneling via Web server (port 80)
• VPN tunneling (Virtual Private Network)
• Servers are located in a public LAN (outside of firewall)
Solutions for the Firewall Problem

- open all ports
- open one port for proxy
- HTTP tunneling via Web server (port 80)
- VPN tunneling (Virtual Private Network)
- Servers are located in a public LAN (outside of firewall)
- connections are established from inside to outside
MOSCITO Solution: combination of different approaches

- open all ports
- open one port for proxy
- HTTP tunneling via Web server (port 80)
- VPN tunneling (Virtual Private Network)
- Servers are located in a public LAN (outside of firewall)
- connections are established from inside to outside
- access control for hosts, domains
Encrypted Data Transfer

MOSCITO provides mechanisms for data encryption based on the JCE (Java Cryptography Extensions).

A Master Key and a dynamically generated Session Key allow encryption of all transferred data.
Usage of Master Key and Session Key

Master Key

Session Key

Master

Users

Agents
Übersicht

✓ MOSCITO –
  Basic Idea, Applications, Advantages

✓ Fundamental Concept and
  Infrastructure

✓ Security Aspects –
  Passing Firewalls and Data Encryption
Summary

MOSCITO has been developed as an Internet-based tool integration platform.

MOSCITO provides many features supporting applications for system design (tool encapsulation and tool coupling, remote simulation) and for e-learning.

MOSCITO software provides open interfaces for further extensions (e.g. direct collaboration between MOSCITO users).

MOSCITO is ready for a migration to Web Services applications using WSDL, UDDI, and SOAP in the future.
Thank you for your attention.