

Software components

Marco Danelutto

Dept. Computer Science

Univ. of Pisa



*Master Degree (Laurea Magistrale) in
Computer Science and Networking
Academic Year 2009-2010*





Introduction

- **Several different definition of components**
- **Kind of common “kernel” of properties agreed**
- **Idea around since the '80**
- **More recently adopted by software companies**
 - *basically to enhance productivity*

The roots

- **reusability**

- *possibility to use X in a context different from the one where it was developed*

- **modularity**

- *X and Y , independently developed, share a common interface that allows interoperability*
- *building blocks for larger sw*

- **compositionality**

- *X and Y can be composed and the composition can be looked at as Z (with no knowledge/perception on the internals)*

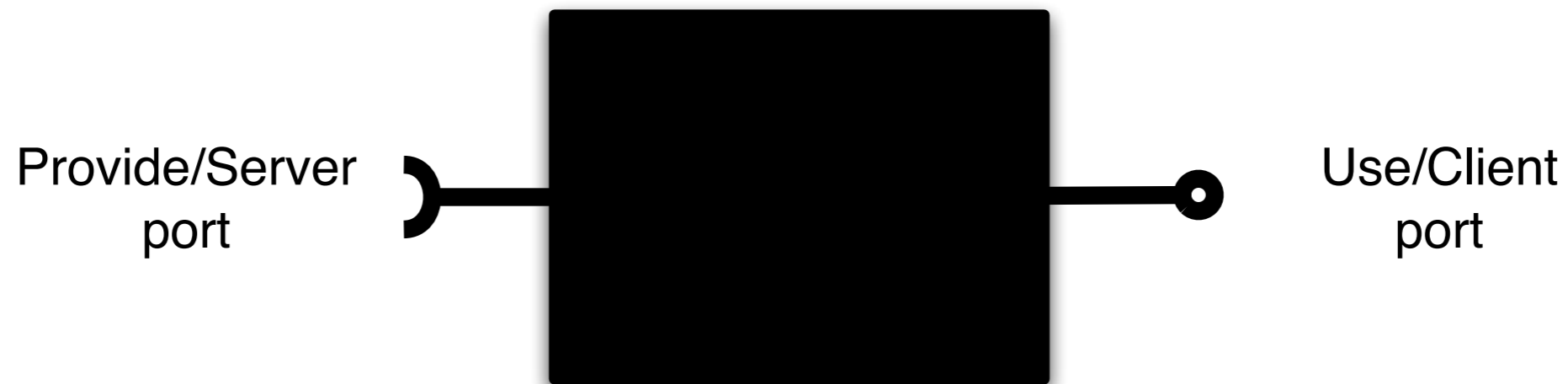
Some definitions

- **reusable software component = logically cohesive, loosely coupled module that denotes a single abstraction (Booch'87)**
- **building blocks from which different software systems can be composed ... plug-compatible in as many ways as possible (Czarnecki & Eisenecker'00)**
- **unit of independent deployment + unit of third-party composition + no (externally) observable state (Szyperski'02)**
- **An individual component is a software package, or a module, that encapsulates a set of related functions (or data). (wikipedia, today)**

Our definition

- **unit of independent deployment and composition**
 - *a component can be deployed on a remote machine*
 - *it can be composed exploiting plugs / ports / interfaces*
- **providing clearly identified service(s)**
 - *provide/server interface ports*
- **requiring clearly identified service(s)**
 - *use/client interface ports*
- **with distinct, possibly collective interaction mechanisms rpc, event, stream, collective (broadcast, multicast, scatter, ...)**

Component



- **Black box**

- *fundamental property to guarantee interoperability, portability, etc.*

- **Provide ports**

- *used to invoke a service of the component*

- **Use ports**

- *used to denote a dependency of the component*

Component composition

- **Use/provide port matching**
- **Usually static**
 - *ASSEMBLY Language (e.g. XML with proper DTD)*
 - ADL: Assembly Description Language

