

# DEPLOYMENT DIAGRAMS

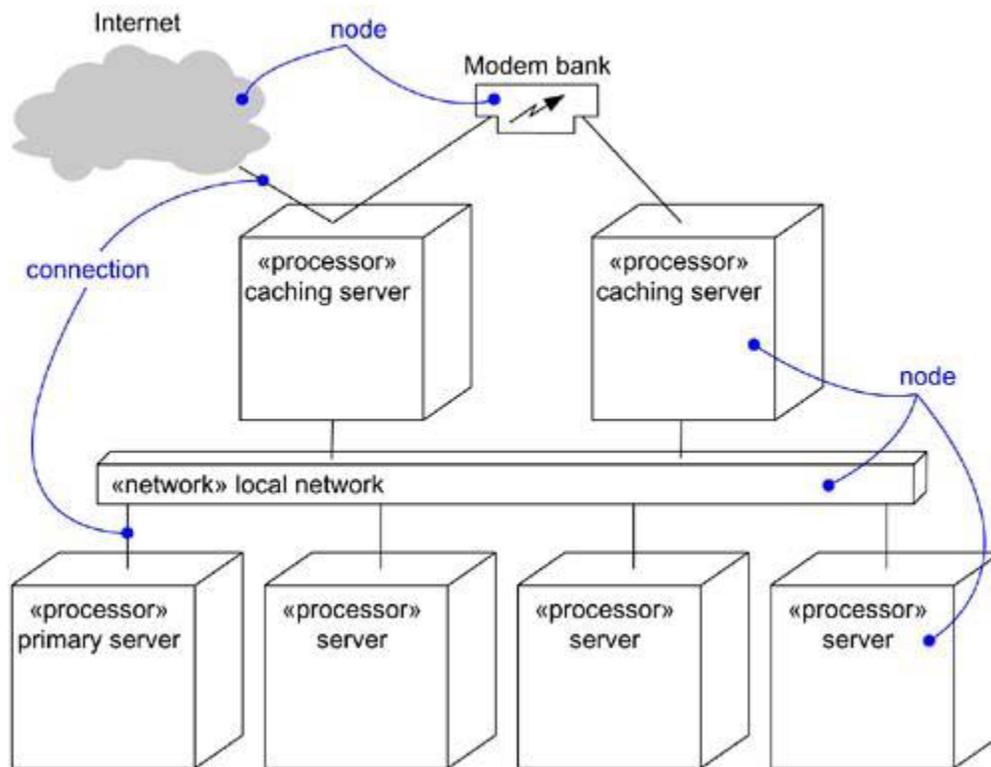


Figure 1: Deployment Diagram

## Deployment Diagrams

- A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them.
- Deployment diagrams are one of the two kinds of diagrams used in modeling the physical aspects of an object-oriented system.
- used to model the static deployment view of a system (topology of the hardware)
- A deployment diagram is just a special kind of class diagram, which focuses on a system's nodes.
- Graphically, a deployment diagram is a collection of vertices and arcs.
- Deployment diagrams commonly contain Nodes and Dependency & association relationships. It may also contain notes and constraints.

- Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering.

### **Common Uses**

Deployment diagrams are used in one of three ways,

- To model embedded systems
- To model client/server systems
- To model fully distributed systems

### **Modeling an Embedded System**

To model an embedded system,

- Identify the devices and nodes that are unique to your system.
- Provide visual cues, especially for unusual devices, by using the UML's extensibility mechanisms to define system-specific stereotypes with appropriate icons. At the very least, you'll want to distinguish processors (which contain software components) and devices (which, at that level of abstraction, don't directly contain software).
- Model the relationships among these processors and devices in a deployment diagram. Similarly, specify the relationship between the components in your system's implementation view and the nodes in your system's deployment view.
- As necessary, expand on any intelligent devices by modeling their structure with a more detailed deployment diagram.

Figure 2 shows the hardware for a simple autonomous robot

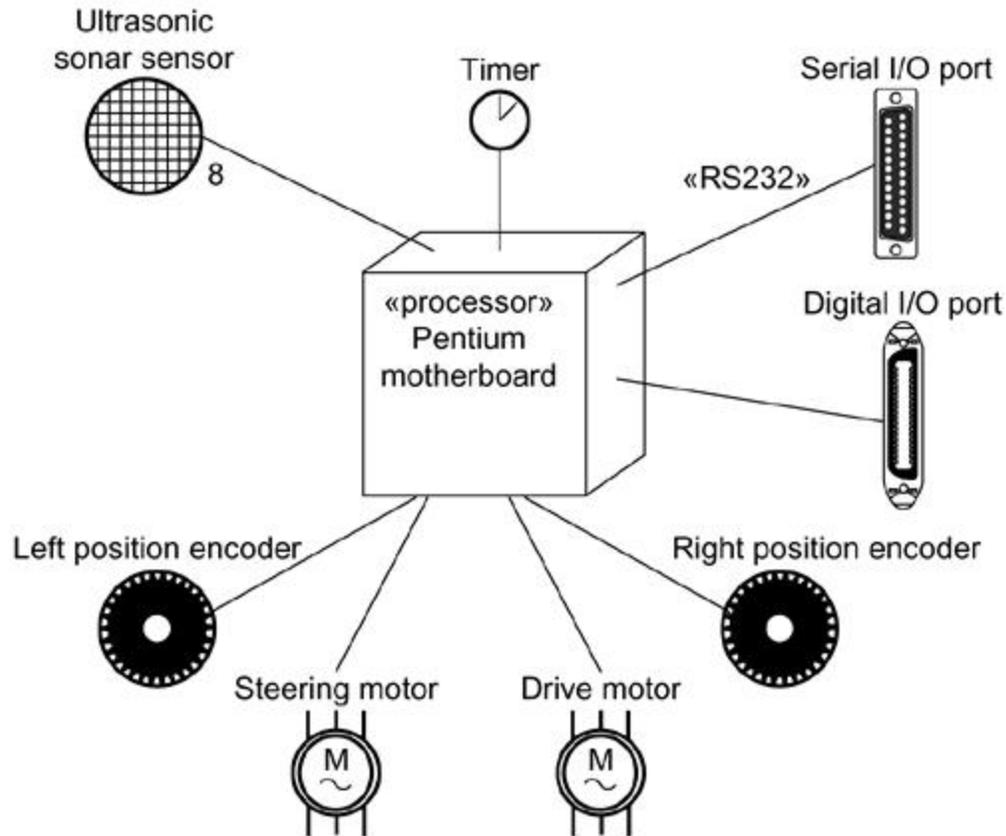


Figure 2: Modeling an Embedded System

### Modeling a Client/Server System

To model a client/server system,

- Identify the nodes that represent your system's client and server processors.
- Highlight those devices that are germane to the behavior of your system. For example, you'll want to model special devices, such as credit card readers, badge readers, and display devices other than monitors, because their placement in the system's hardware topology are likely to be architecturally significant.
- Provide visual cues for these processors and devices via stereotyping.
- Model the topology of these nodes in a deployment diagram. Similarly, specify the relationship between the components in your system's implementation view and the nodes in your system's deployment view.

Figure 3 shows the topology of a human resources system, which follows a classical client/server architecture.

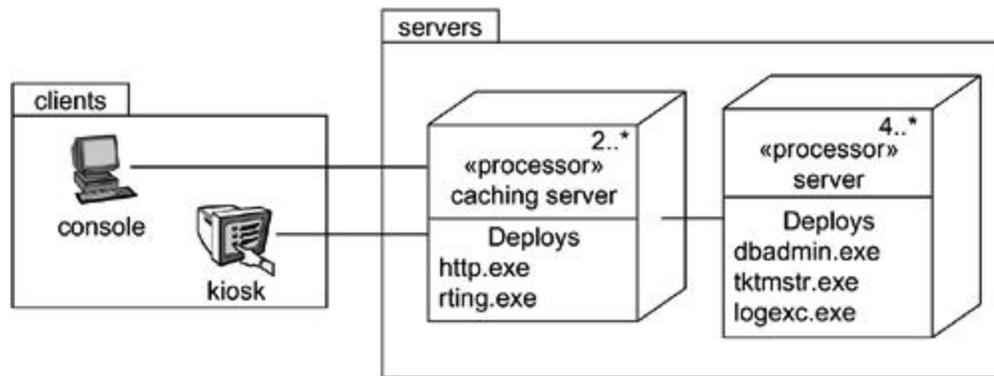


Figure 3: Modeling a Client/ Server System

### Modeling a Fully Distributed System

To model a fully distributed system,

Identify the system's devices and processors as for simpler client/server systems.

If you need to reason about the performance of the system's network or the impact of changes to the network, be sure to model these communication devices to the level of detail sufficient to make these assessments.

Pay close attention to logical groupings of nodes, which you can specify by using packages.

Model these devices and processors using deployment diagrams. Where possible, use tools that discover the topology of your system by walking your system's network.

If you need to focus on the dynamics of your system, introduce use case diagrams to specify the kinds of behavior you are interested in, and expand on these use cases with interaction diagrams.

When modeling a fully distributed system, it's common to reify the network itself as an node. eg:- Internet, LAN, WAN as nodes

Figure 4 shows the topology of a fully distributed system.

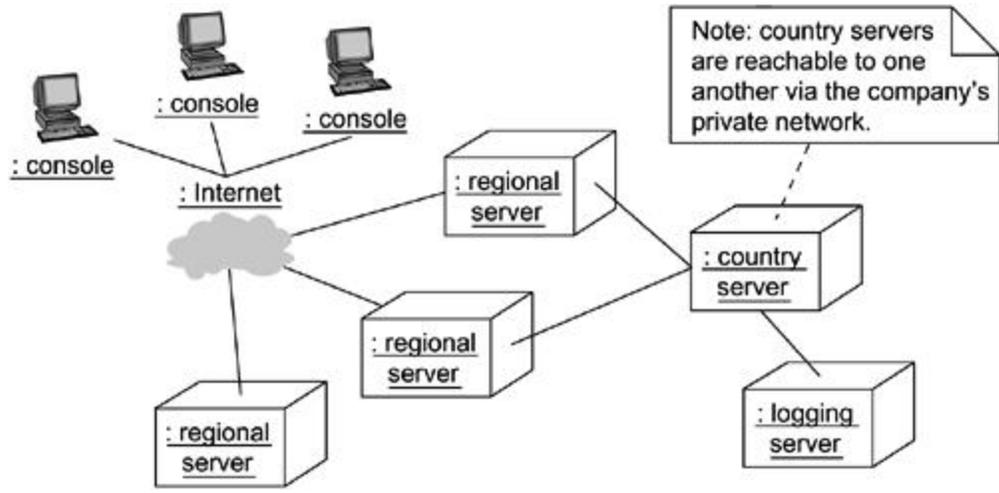


Figure 4: Modeling a Fully Distributed System

Source : <http://praveenthomasln.wordpress.com/2012/04/07/deployment-diagrams-s8-cs/>