CHAPTER 14

Deployment Diagrams

Figure 14-1: Environment Model View—Deployment Diagrams

Deployment diagrams (Figure 14-1) render the environment view of a system. These diagrams describe the configuration of processing resource elements and the mapping of software implementation components onto them. These diagrams contain nodes, components, and their relationships. They may be constructed using the following technique:

1. Model nodes.
2. Model communication relationships between nodes.
3. Model components (from component diagrams).
4. Model run-time relationships between components.
5. Model calls relationships between components (from component diagrams).
6. Model supports relationships between nodes and components.
7. Model becomes relationships between components.
8. Refine and elaborate as required.

Deployment diagrams may have the various expressions (conditions, etc.) they utilize be expressed using pseudocode or another language.

Nodes

Nodes (Figure 14-2) are classes that define run-time physical objects. They are used to model processing or computational resources. Nonhuman resources generally have at least a memory and often processing capability. Human resources are people.

Figure 14-2: Nodes

Nodes

- Are denoted as three-dimensional cubes.
- Have types that represent different kinds of computational resources. Node types must have a name or identifier string that represents the type of the node.
- Have instances that represent specific computational resources. Node instances
  - Must have a name or identifier string that represents the name of the node. The name must be underlined.
  - May have a colon followed by a type expression that indicates the type of the node.
- May have aggregation relationships with other components, nodes, objects, and processes (or active objects), indicating that instances reside on nodes or are deployed onto nodes. The "location" tagged value may be used to specify this relationship (Chapter 15).
- May use other notation of classes and objects.

Communication Relationships

Communication relationships (Figure 14-3) are associations between nodes. They are used to model communication paths or connections between nodes.

Figure 14-3: Communication Relationship

Communication relationships

- Are denoted as solid associations.
- May be stereotyped to indicate the nature of the communication path or connection.
- May use other notation of associations and links.

Run-Time Relationships

Run-time relationships (Figure 14-4) are associations between components. They are used to model execution dependencies, that is, the dependency of one component on another component during run time.

Figure 14-4: Run-Time Relationships

Run-time relationships

- Are denoted as dashed arrows.
- Have components with the following properties:
  - Client components are dependent on supplier components.
  - Supplier elements may be objects.
  - Supplier components exist at run time but need not exist at development time.
- May be stereotyped to depict an implementation-specific run-time dependency. When not stereotyped, the relationship represents a run-time communication or calls relationship.

Supports Relationships

Supports relationships (Figure 14-5) are associations between nodes and components or objects. They are used to model a component's or object's ability to execute on a node or a node's capability to host a component or object.

Supports relationships

- Are denoted as dashed arrows.

Figure 14-5: Supports Relationship

- May be stereotyped using the “supports” keyword. When no stereotype keyword is used, the “supports” keyword is assumed.
- May be more precisely stereotyped to depict an implementation-specific dependency.
- May be specified using the “location” property (tagged value) of a component or object.

Becomes Relationships

Becomes relationships (Figure 14-6) are associations between components or objects. They are used to model the migration of components among nodes or the migration of objects among components or nodes.

Figure 14-6: Becomes Relationships

Becomes relationships

- Are denoted as dashed arrows.
- Must be stereotyped using the “becomes” keyword.
- May have a time property attached to indicate the time at which the movement or migration occurs. Messages are directed to the correct occurrence of the component.