
Component-Based Software Engineering in a Pervasive Environment

David Garlan & Bradley Schmerl
Carnegie Mellon University

*4th ICSE Workshop on Component-Based Software Engineering
14-15 May 2001*

Thesis

- **Component-based SE is maturing as a discipline**
 - > Books, Conferences, Handbooks, Tools, Languages
- **But technology does not stand still**
 - > Moore's Law and the Internet are causing drastic changes in the use of software systems and requirements for new systems
- **A critical new area of concern is ubiquitous and pervasive computing**
 - > Universal connectivity
 - > Proliferation of devices and information sources
 - > "Computers" everywhere
 - > Mobile computing
- **Requires reinterpreting ideas for component-based computing**

From Workstations to Pervasive Computing

Today

- Isolated workstations
- Applications
- Files
- Manual invocation of computation
- Manual configuration of computing environment
- Manual coordination between different computing environments
- Impoverished UI capabilities (WIMPy)
- Poor connectivity

Tomorrow

- Computing “rooms”
- Tasks
- Information
- Speech- and gesture-mediated invocation
- Automatic tailoring of services to environment
- Transparent support for moving tasks to new situations
- Rich interface capabilities
- Universal connectivity

Example Scenario

- Fred is in his office, frantically preparing for a meeting with a presentation and software demonstration across campus.
- It is time to leave, but Fred is not quite ready. He grabs his PalmXXII wireless handheld computer and walks out of the door.
- The system infrastructure transfers the state of his work to his handheld, and allows him to make his final edits using voice commands during his walk.
- The system infers where Fred is going from his calendar and the campus location tracking service.
- It downloads the presentation and demonstration software to the projection computer, and warms up the projector.

Necessary ingredients of a solution

Achieving this task requires addressing four concerns:

1. Mobility:

- > As people move from one environment to another, they will expect their tasks to logically “follow” them around so that they can continue working

2. Adaptability:

- > Computing should be able to adapt to accommodate resources that dynamically enter/leave the environment, and changing user tasks/priorities

3. Resource awareness:

- > Pervasive software should make optimal use of the available resources to support user tasks

4. Representation of tasks (not covered in this talk)

New interpretations of good SE ideas

1. Reduce complexity through abstraction (information hiding)

- > Today: Hide implementation details
- > Tomorrow: Expose resource usage properties

2. Correct computation is the primary concern

- > Today: Computation is the independent variable, and determines the resources needed
- > Tomorrow: Resource availability may determine the fidelity of the computation

3. Systems should satisfy the user requirements

- > Today: Requirements (and hence correctness) can be determined in advance
- > Tomorrow: Requirements (and hence correctness) depend on a user's task and environment

New interpretations of good ideas (cont)

4. Keep errors from propagating to the user

- > Today: Write correct software
- > Tomorrow: System is self-monitoring, implying that components give information about their health

5. Design systems with evolution in mind

- > Today: Configuration management techniques to build systems for predetermined environment(s)
- > Tomorrow: Components self-configure and self-tune for different environments as they are encountered

Implications for Component Engineering

• Components should:

- > Expose resource requirements and usage
- > Provide a monitoring interface
- > Support multi-fidelity computation

• Component-based systems should:

- > Provide mechanisms for system adaptation
- > Support utility-based component composition
- > Include mechanisms for understanding the health of the system
- > Systems should have some knowledge/understanding of user intent