Overview and Basic Concepts

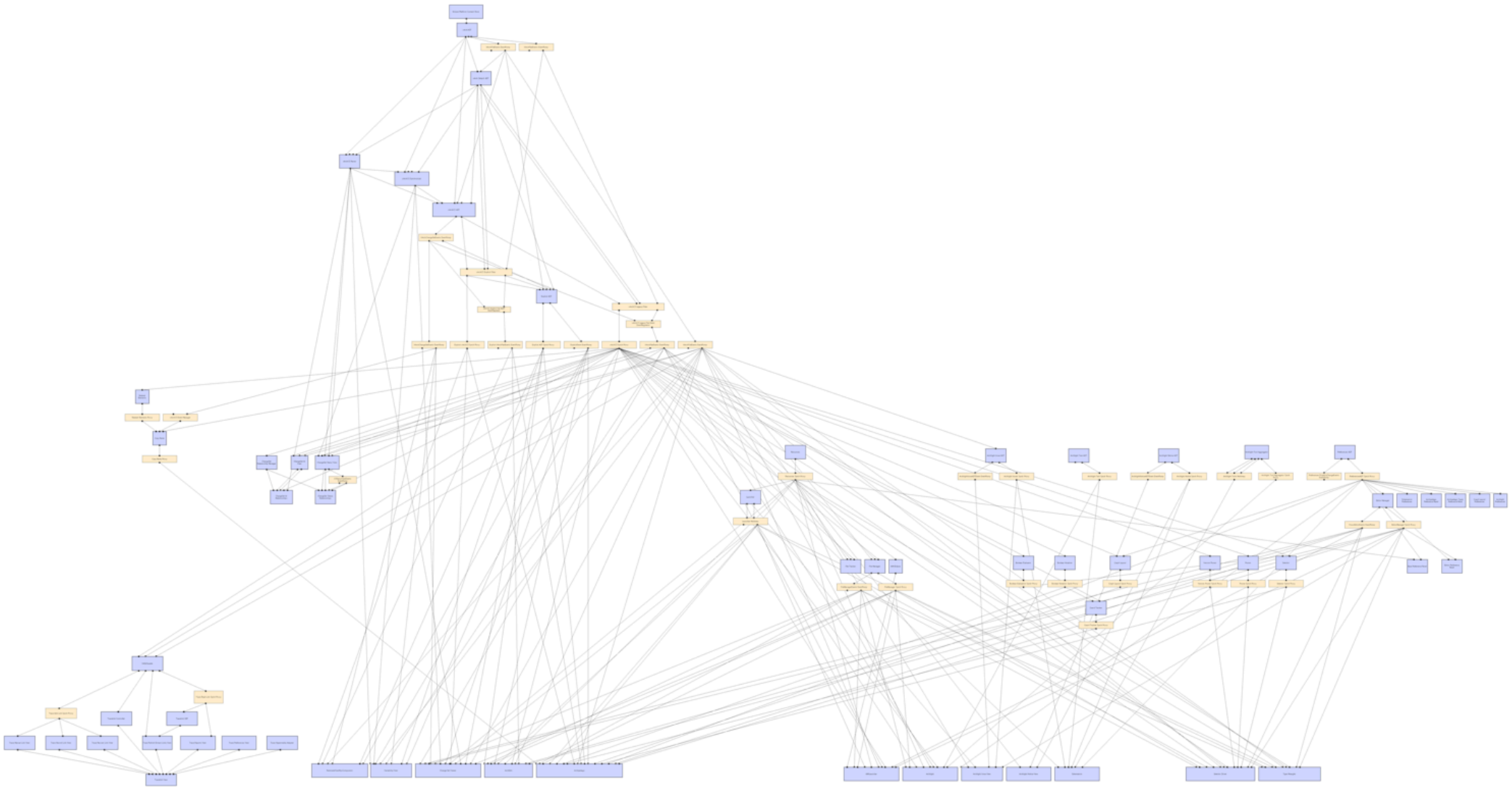
Instructor: Yongjie Zheng
August 24, 2015

Software Engineering

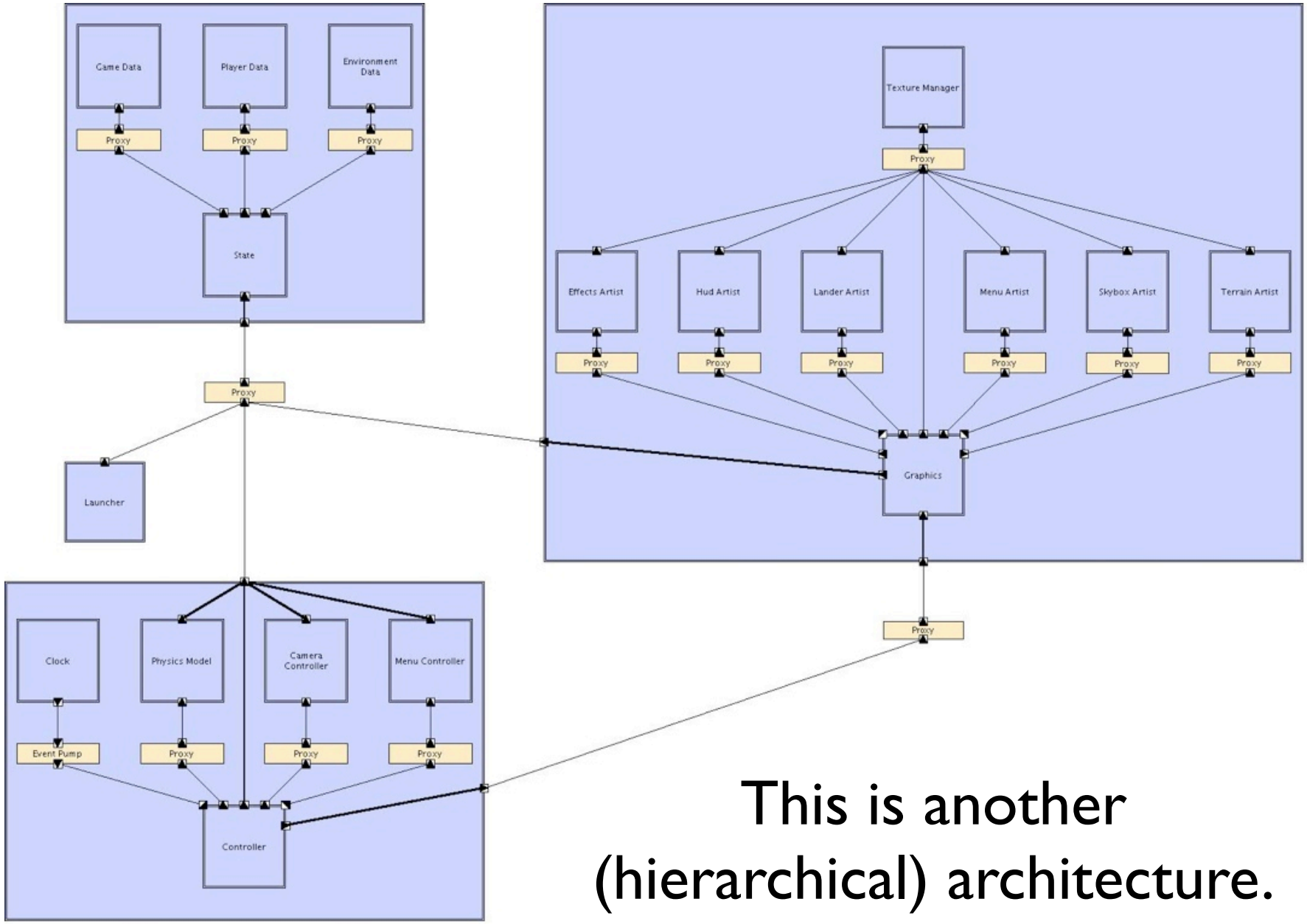
- Software engineering is about improving predictability, productivity, and quality of software production.
- Software engineering is different from other engineering disciplines primarily because
  - Software has four essential difficulties [Fred Brooks]: complexity, conformity, changeability, and invisibility.
  - The tools that people use to build software usually are software, too.
- Software engineering includes the following areas (not limited to): software process, requirements engineering, software architecture and design, software testing, ...

Software Architecture

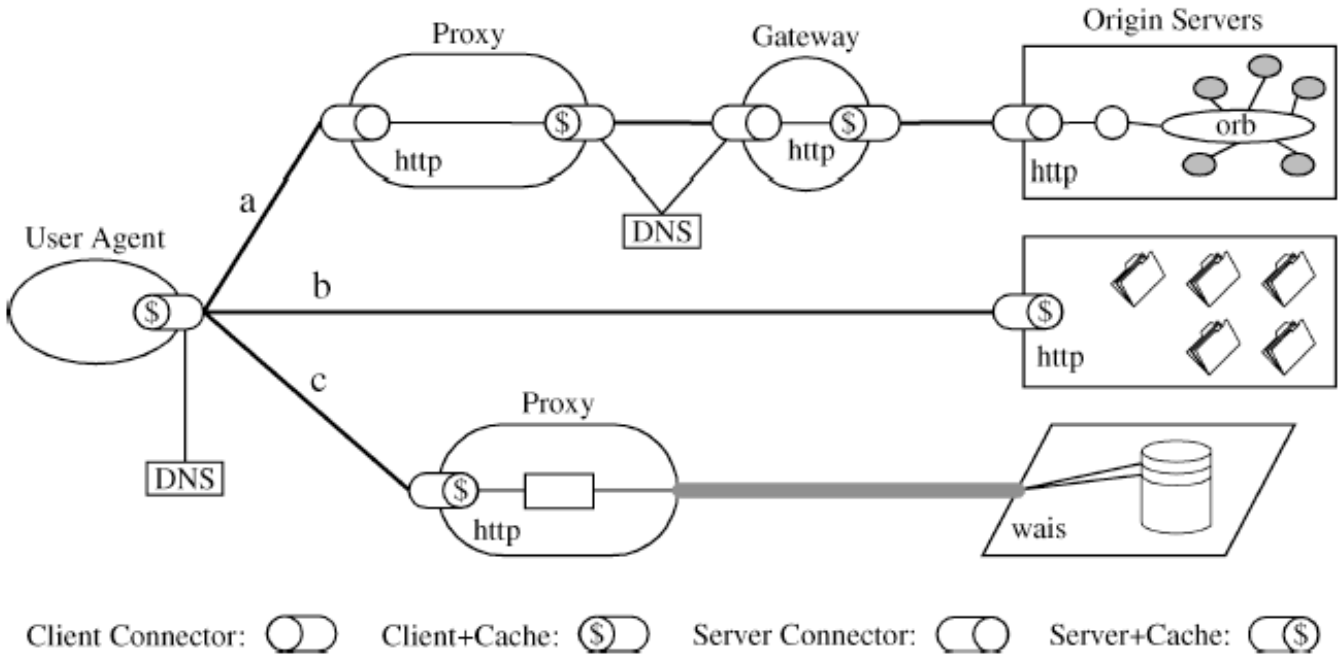
- The definition used in traditional software engineering: the top-level decomposition of a software system.
- The definition widely adopted in the software architecture community: the description of elements from which systems are built, interactions among those elements, patterns that guide their composition, and constraints on these patterns.
  - 4C model: Component, Connector, Configuration, and Constraint.
- The definition given in our text book: a software system's architecture is the set of principal design decisions made about the system.



This is an architecture.



This is another (hierarchical) architecture.



This is still an architecture.

Benefits of Software Architecture

- High-level abstraction: closer to the problem domain compared with programming languages.
  - Facilitates the communication with stakeholders.
- High-level reuse: reuse source code and architecture.
  - Architecture styles and patterns.
- Early system analysis: the earlier a problem is found in software development, the cheaper it is to fix the problem.
- Supports software maintenance.
  - Helps software maintainers (who are usually not the original developers) understand the system.
  - Helps the estimation of evolution cost.

Architectures in Action

- Pipes and filters in Unix shell
  - Example: ls -l | grep "5590SA" | more
  - Pipe: the symbol "|"; Filter: grep, more, ...
- The MVC pattern in user interface design.
  - MVC: Model-View-Controller
- Domain-specific software architecture in product-line applications
  - Product line Hall of Fame (http://splc.net/fame.html)
- The REST style in WWW
  - REST: REpresentational State Transfer

Software Architecture Research

- Conceptualized from 1980's: module interconnection language (MIL), programming-in-the-large, ...
- Formal definitions were introduced in 1992/1993, around when an independent research area was formed.
- Institutions: Carnegie Mellon University, Imperial College London, University of Washington Seattle, University of California Irvine, ...
- Conferences: International Conference on Software Engineering (ICSE), International Symposium on the Foundations of Software Engineering (FSE), WICSA, ...

Topics of Software Architecture

- Architecture styles
- Architecture description languages
- Architecture frameworks
- Product line architectures / domain-specific software architecture
- Architecture recovery
- Architecture of network and distributed systems
- Architecture analysis and simulation
- Architecture-based ... (e.g. adaptation, traceability, configuration management)

Basic Concepts

- Software architecture
- Architecture components
- Interfaces
- Architecture connectors
- Architecture configurations
- More concepts (e.g. architecture styles, architecture drift and erosion, architecture views) will be introduced when we go through specific topics.

Revising Definition of Software Architecture

- The set of principal design decisions made about a software system.
- "Design decisions" encompass every aspect of the system under development, including
  - System structure
  - Functional behavior
  - Interaction among system elements
  - Nonfunctional properties
- How "principal" is defined is depending on the system goals, and is subject to the interpretation of the system's stakeholders.

Architecture Components

- An architecture component is an architectural entity that
  - Encapsulates a subset of the system's functionality and/or data;
  - Restricts access to that subset via an explicitly defined interface;
  - Has explicitly defined dependencies on its required execution context.
- An architecture component is a locus of computation and state in a system, and typically provides application-specific services.

Interfaces

- An architecture component normally has multiple provided interfaces and required interfaces.
  - An interface is a specification of component behavior and properties; technically, an interface is a set of named operations that can be invoked by clients.
  - A provided interface provides a service as a contract for other components to use.
  - A required interface represents what the external environment needs to provide for the component to function.

Architecture Connectors

- An architecture connector is an architectural element tasked with effecting and regulating interactions among components.
- Connectors typically provide application-independent interaction facilities.
- In many software systems connectors are usually simple procedure calls or shared data accesses.
- Much more sophisticated and complex connectors are possible!
  - Streaming connectors, distribution connectors, wrapper/adaptor connectors.

Architecture Configuration

- An architectural configuration is a set of specific associations between the components and connectors of a software system's architecture.