COMP 6471
Software Design Methodologies

Fall 2011

Dr Greg Butler

http://www.cs.concordia.ca/~gregb/home/comp6471-fall2011.html

Software Architecture Document

Architectural Representation

(Summary of how the architecture will be described in this document, such as using by technical memos and the architectural views. This is useful for someone unfamiliar with the idea of technical memos or views. Note that not all views are necessary.)

Architectural Factors

(Reference to the Supplementary Specification to view the Factor Table.)

Architectural Decisions

(The set of technical memos that summarize the decisions.)

Logical View

(UML package diagrams, and class diagrams of major elements. Commentary on the large scale structure and functionality of major components.)

Deployment View

(UML deployment diagrams showing the nodes and allocation of processes and components. Commentary on the networking.)

Process View

(UML class and interaction diagrams illustrating the processes and threads of the system. Group this by threads and processes that interact. Comment on how the interprocess communication works (e.g., by Java RMI).

Use-Case View

(Brief summary of the most architecturally significant use cases. UML interaction diagrams for some architectural significant use-case realizations, or scenarios, with commentary on the diagrams explaining how they illustrate the major architectural elements.)

Other Views...

Logical View

- Conceptual organization of software
  - Layers, subsystems, packages, frameworks, classes, interfaces
  - Summarize functionality of major software elements, eg each subsystem
- Show use-case realization scenarios as interaction diagrams for key aspects of system
- UML package, class, interaction diagrams

Process View

- Shows processes and threads
  - Responsibilities, collaborations
  - Allocation of logical elements (layers, subsystems, classes, ...) to them
  -
- UML class and interaction diagrams
  - Use UML process and thread notation

Deployment View

- Show physical deployment of processes and components to processing nodes
- Show physical network configuration of nodes
- UML deployment diagram

Implementation View

- Summary description of noteworthy organization of
  - Deliverables, eg source code, executables
  - Things that create deliverables, eg scripts, graphics
- Use text and UML package and component diagrams

Data View

- Overview of
  - data flows
  - persistent data schema
  - schema mapping from objects to persistent data
  - mechanism mapping objects to DB
  - stored procedures and triggers

- UML class diagram for data models
- UML activity diagrams for data flows

How many views?

- Simplified models to fit the context
- Not all systems require all views:
    - Single processor: drop deployment view
    - Single process: drop process view
    - Very Small program: drop implementation view
- Adding views:
    - Data view, security view

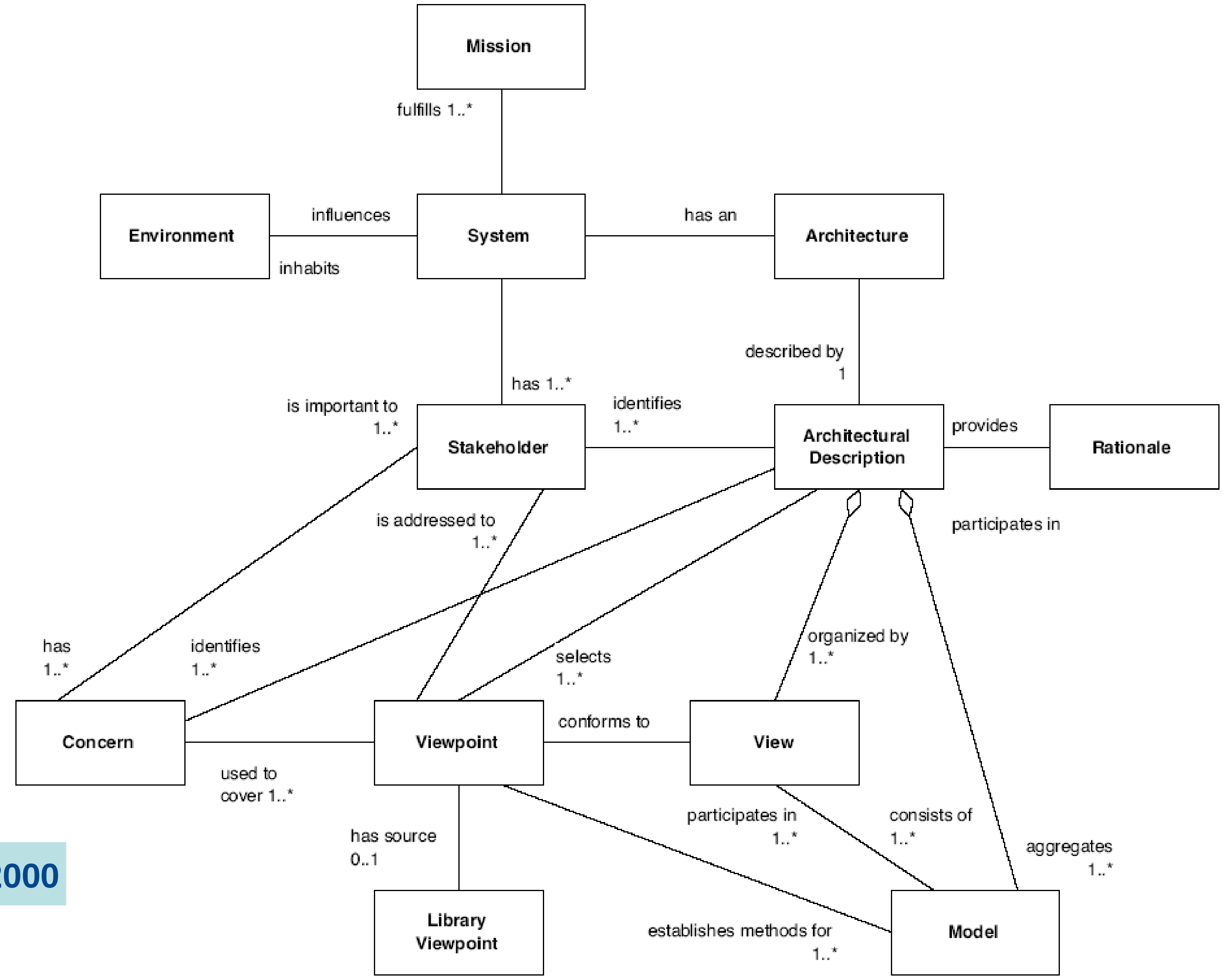
Many stakeholders, many views

- Architecture is many things to many different interested parties
  - end-user
  - customer
  - project manager
  - system engineer
  - developer
  - architect
  - maintainer
  - other developers
- Multidimensional reality
- Multiple stakeholders
  - multiple views, multiple blueprints

Models

- Models are the language of designer, in many disciplines
- Models are representations of the system to-be-built or as-built
- Models are vehicle for communications with various stakeholders
- Visual models, blueprints
- Scale
- Models allow reasoning about some characteristic of the real system

Software Architecture Documentation Conceptual Model



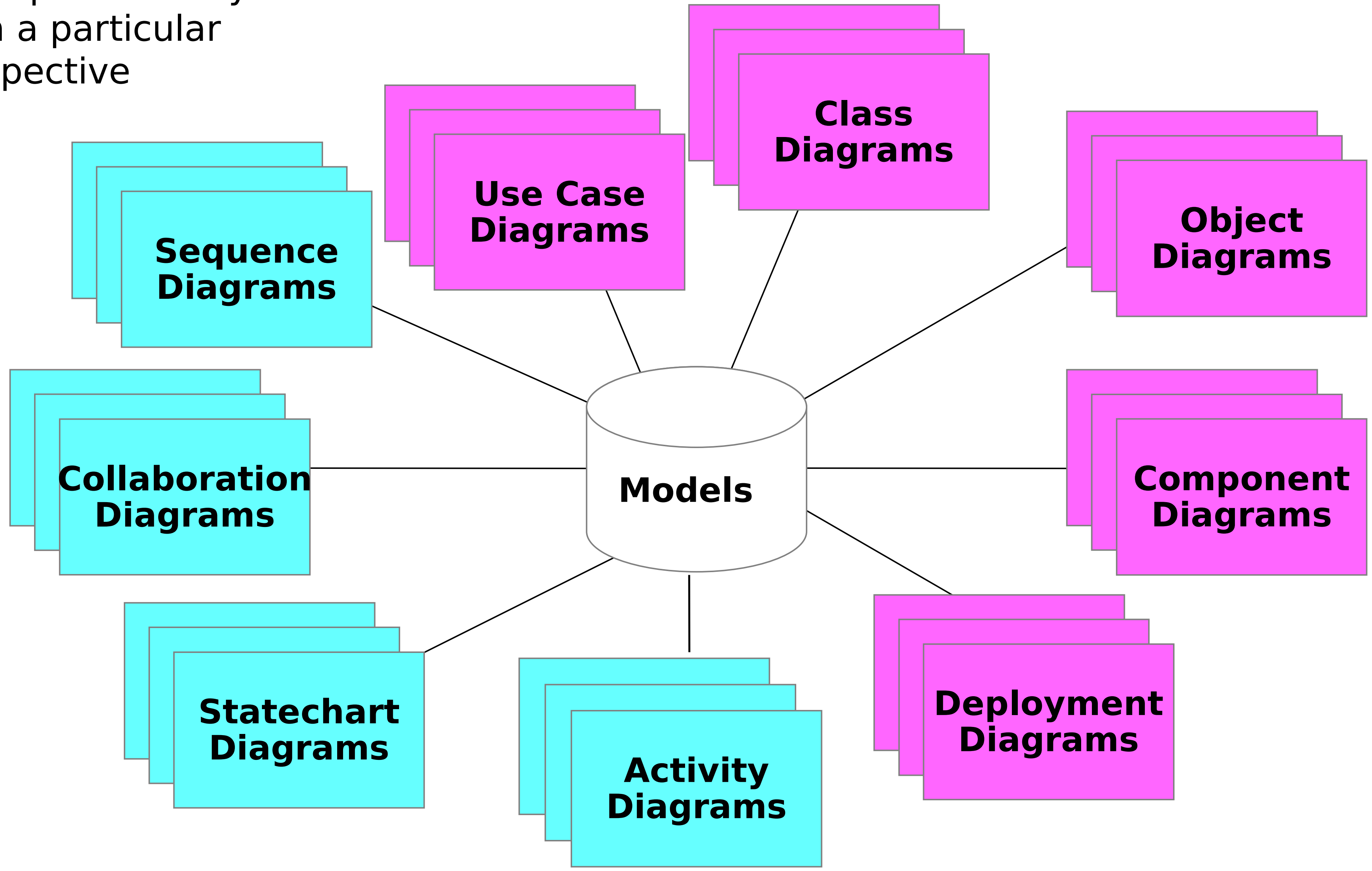
IEEE 1471-2000

Architectural view

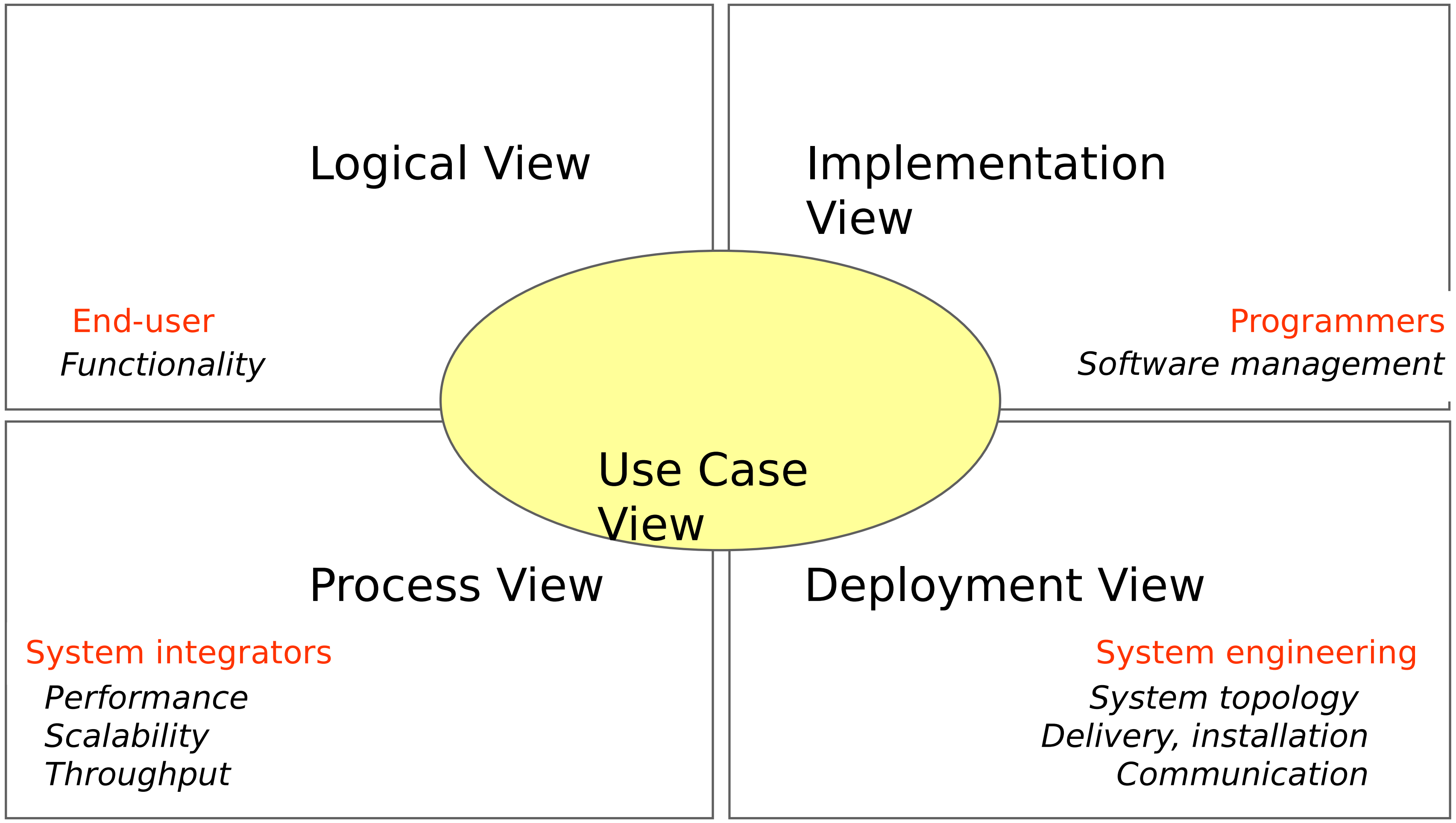
- An architectural view is a simplified description (an abstraction) of a system from a particular perspective or vantage point, covering particular concerns, and omitting entities that are not relevant to this perspective

Models, Views, and Diagrams

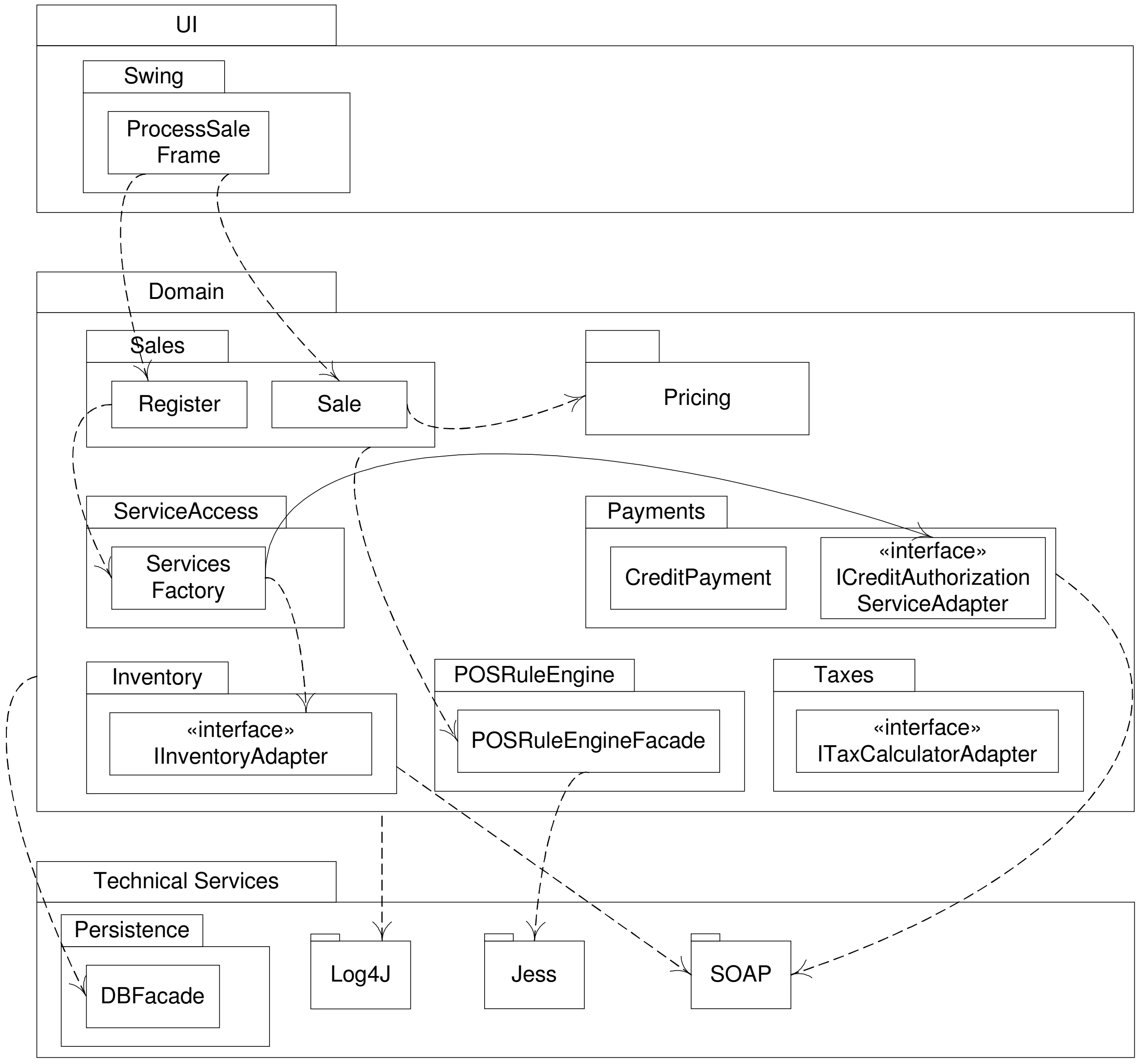
A ***model*** is a complete description of a system from a particular perspective



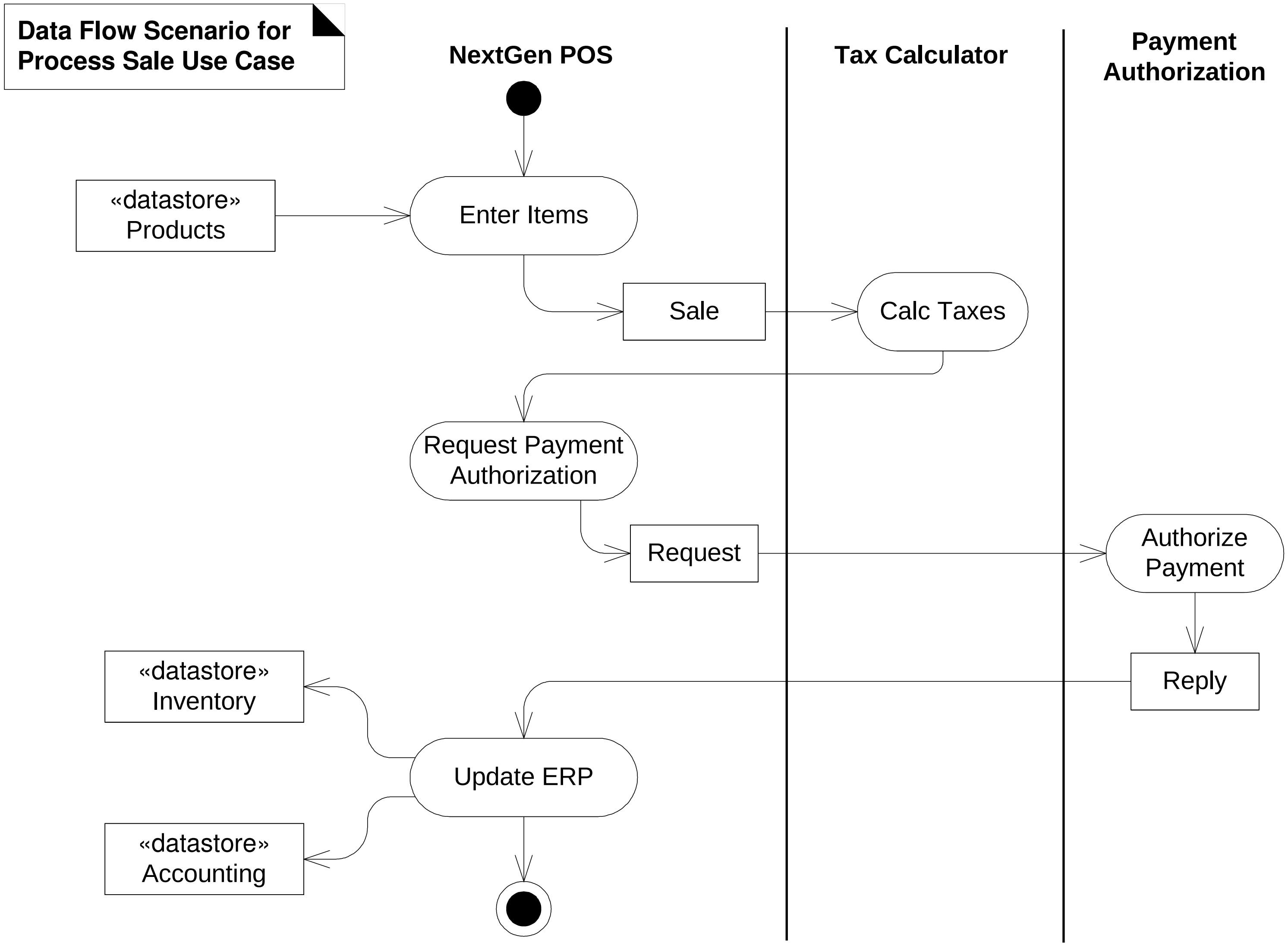
Representing System Architecture
Kruchten's 4+1 View Model



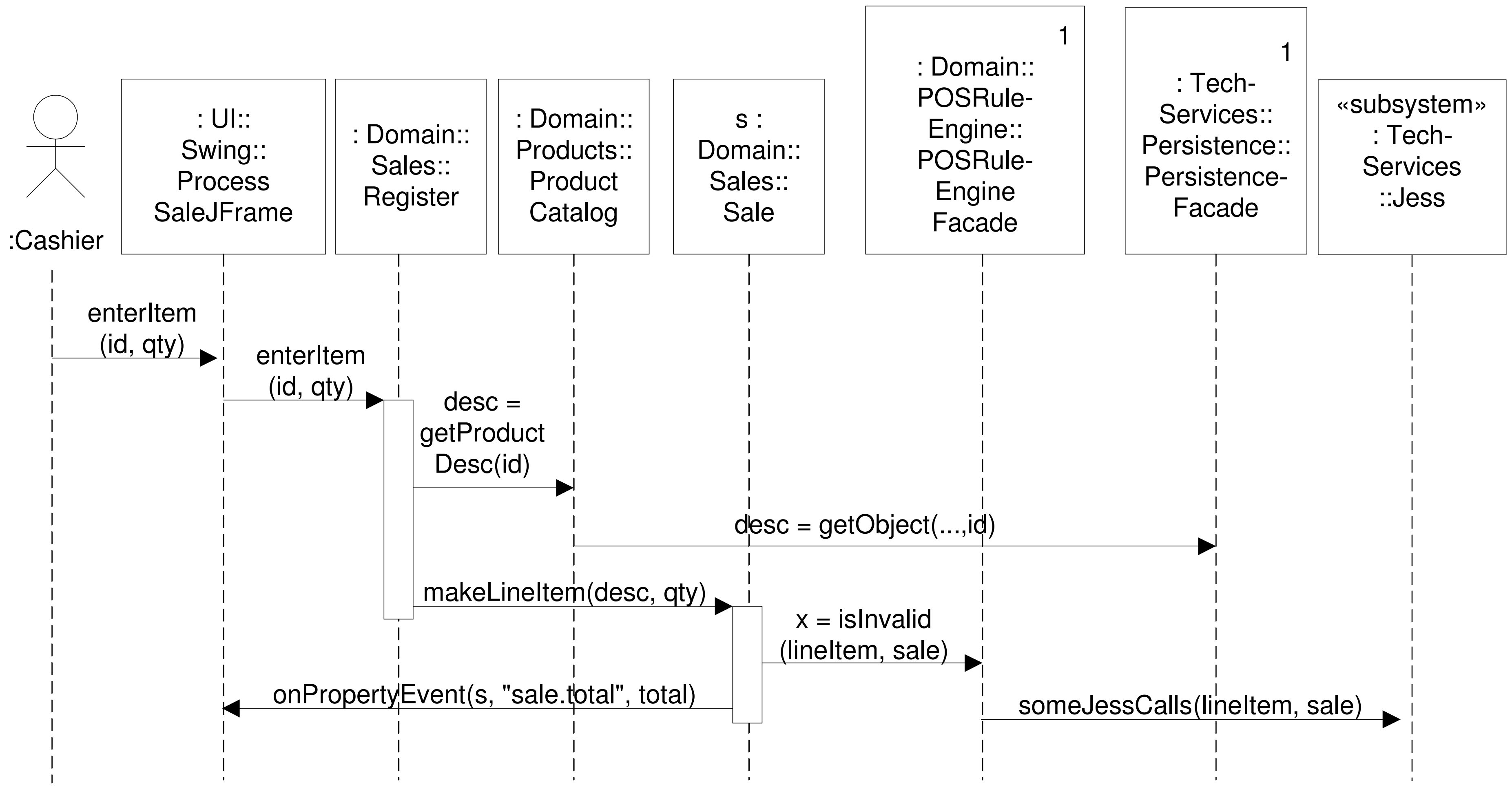
Logical View: Package Diagram



Activity Diagram: Process Model

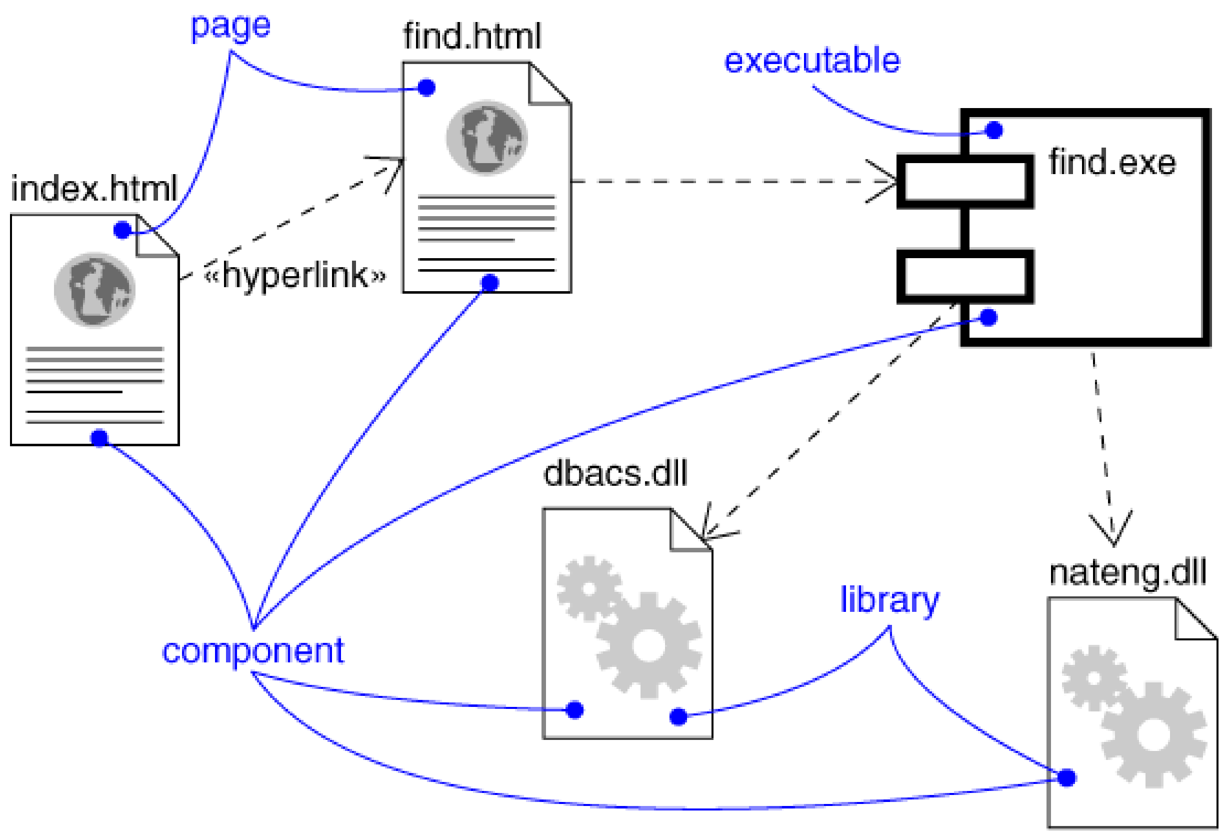


Interaction Diagram: Process Model


:Cashier
: UI:: Swing:: Process SaleJFrame
: Domain:: Sales:: Register
: Domain:: Products:: Product Catalog
s : Domain:: Sales:: Sale
1
: Domain:: POSRule- Engine:: POSRule- Engine Facade
1
: Tech- Services:: Persistence:: Persistence- Facade
«subsystem» : Tech- Services ::Jess
enterItem (id, qty)
enterItem (id, qty)
desc = getProduct Desc(id)
desc = getObject(...,id)
makeLineItem(desc, qty)
x = isInvalid (lineItem, sale)
onPropertyEvent(s, "sale.total", total)
someJessCalls(lineItem, sale)

Component Diagram

- Captures the physical structure of the implementation



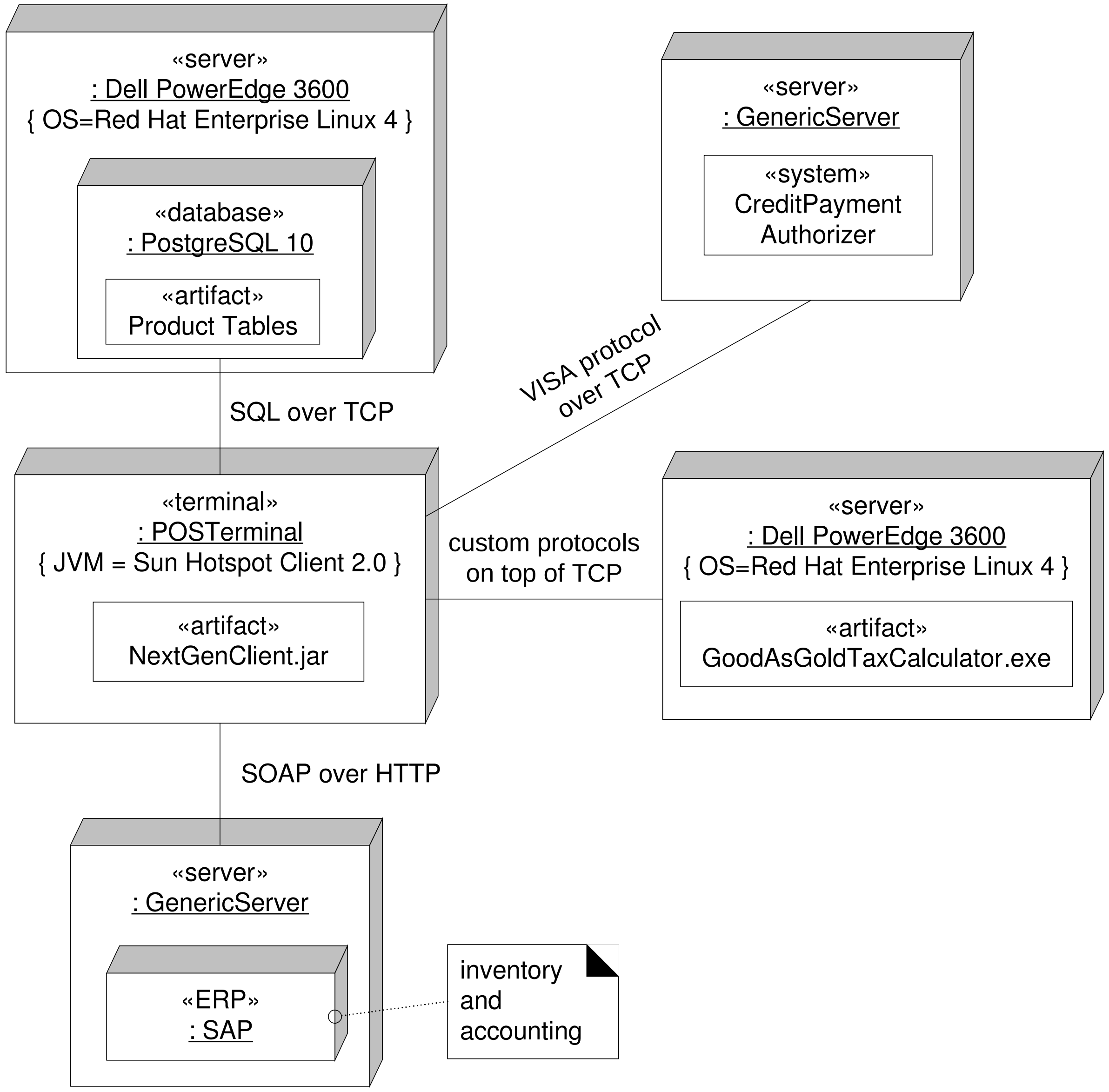
Component Diagram

- Captures the physical structure of the implementation
- Built as part of architectural specification
- Purpose
  - Organize source code
  - Construct an executable release
  - Specify a physical database
- Developed by architects and programmers

Deployment Diagram

- ➢ Captures the topology of a system’s hardware
- ➢ Built as part of architectural specification
- ➢ Purpose
  - Specify the distribution of components
  - Identify performance bottlenecks
- ➢ Developed by architects, networking engineers, and system engineers

Deployment Diagram



Software Architecture Document

Architectural Representation

(Summary of how the architecture will be described in this document, such as using by technical memos and the architectural views. This is useful for someone unfamiliar with the idea of technical memos or views. Note that not all views are necessary.)

Architectural Factors

(Reference to the Supplementary Specification to view the Factor Table.)

Architectural Decisions

(The set of technical memos that summarize the decisions.)

Logical View

(UML package diagrams, and class diagrams of major elements. Commentary on the large scale structure and functionality of major components.)

Deployment View

(UML deployment diagrams showing the nodes and allocation of processes and components. Commentary on the networking.)

Process View

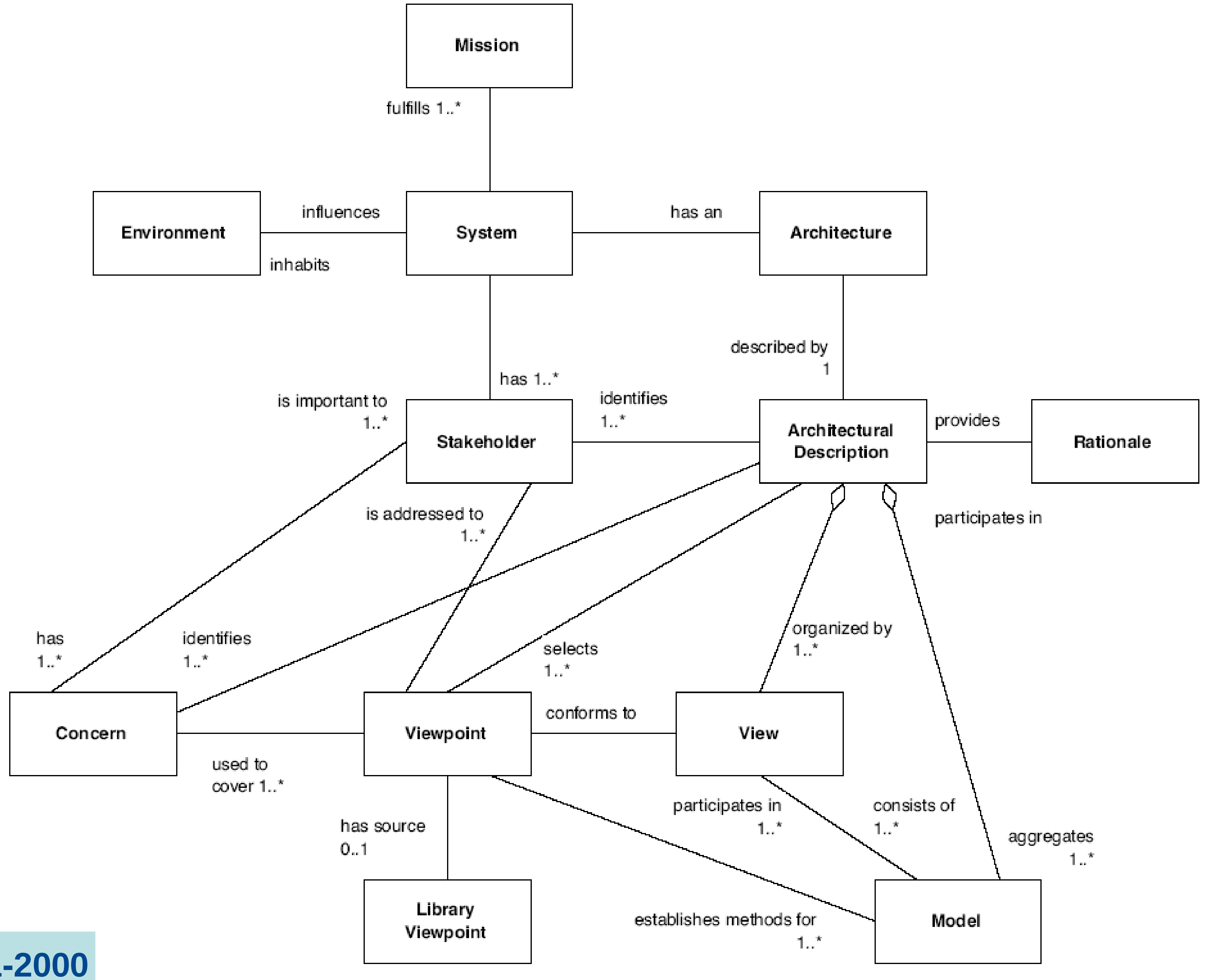
(UML class and interaction diagrams illustrating the processes and threads of the system. Group this by threads and processes that interact. Comment on how the interprocess communication works (e.g., by Java RMI).

Use-Case View

(Brief summary of the most architecturally significant use cases. UML interaction diagrams for some architectural significant use-case realizations, or scenarios, with commentary on the diagrams explaining how they illustrate the major architectural elements.)

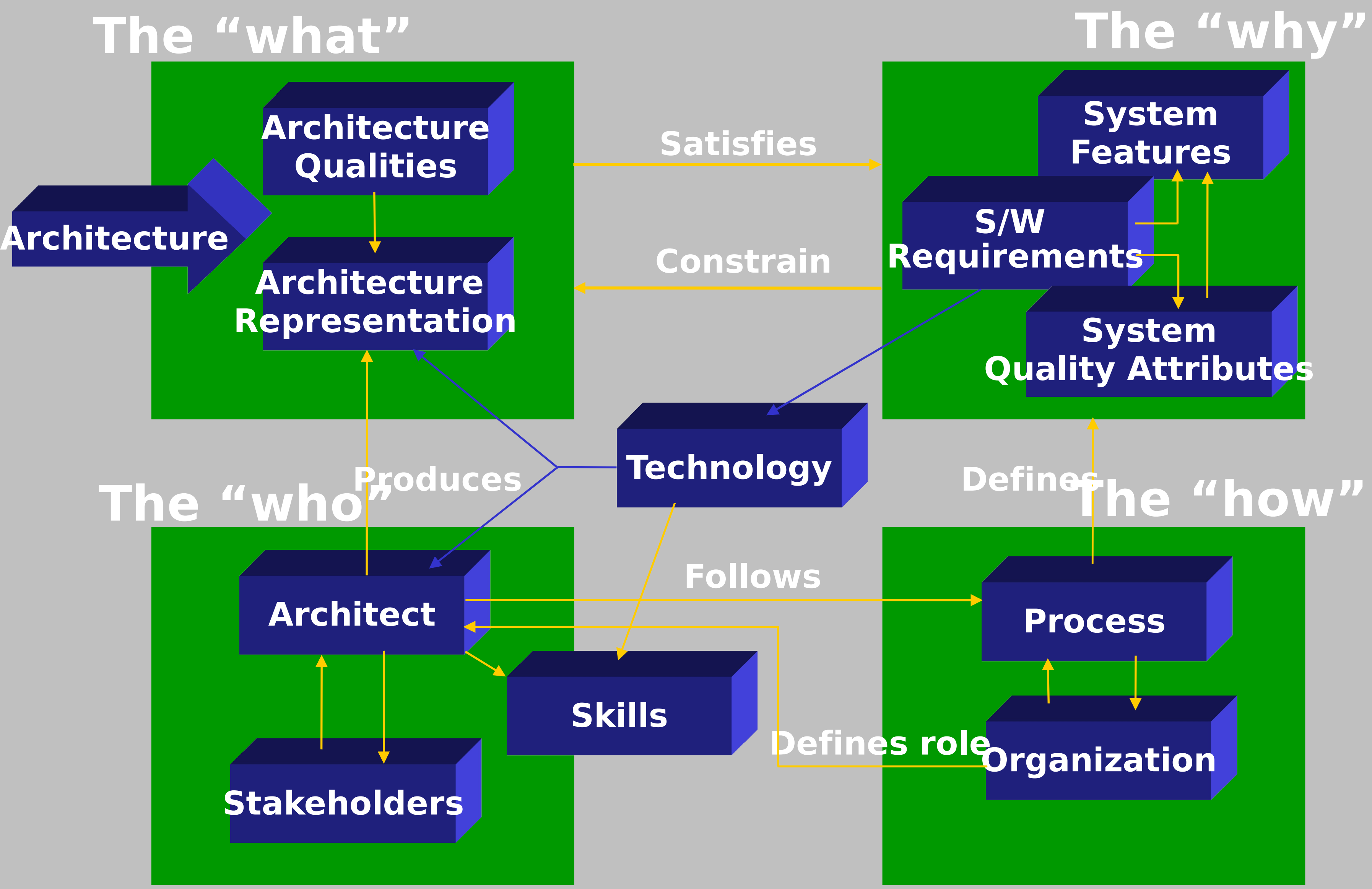
Other Views...

Documentation Conceptual Model

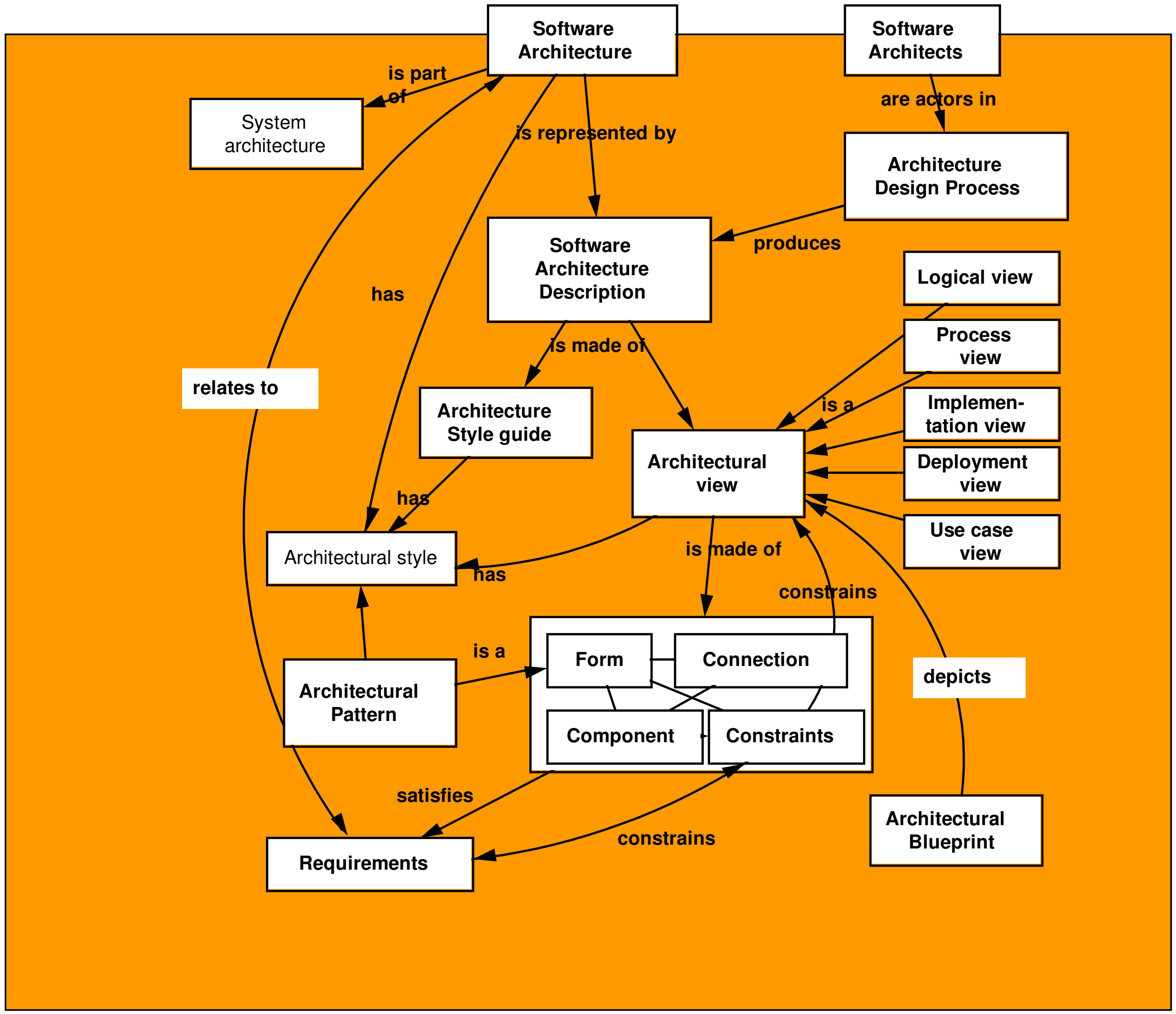


IEEE 1471-2000

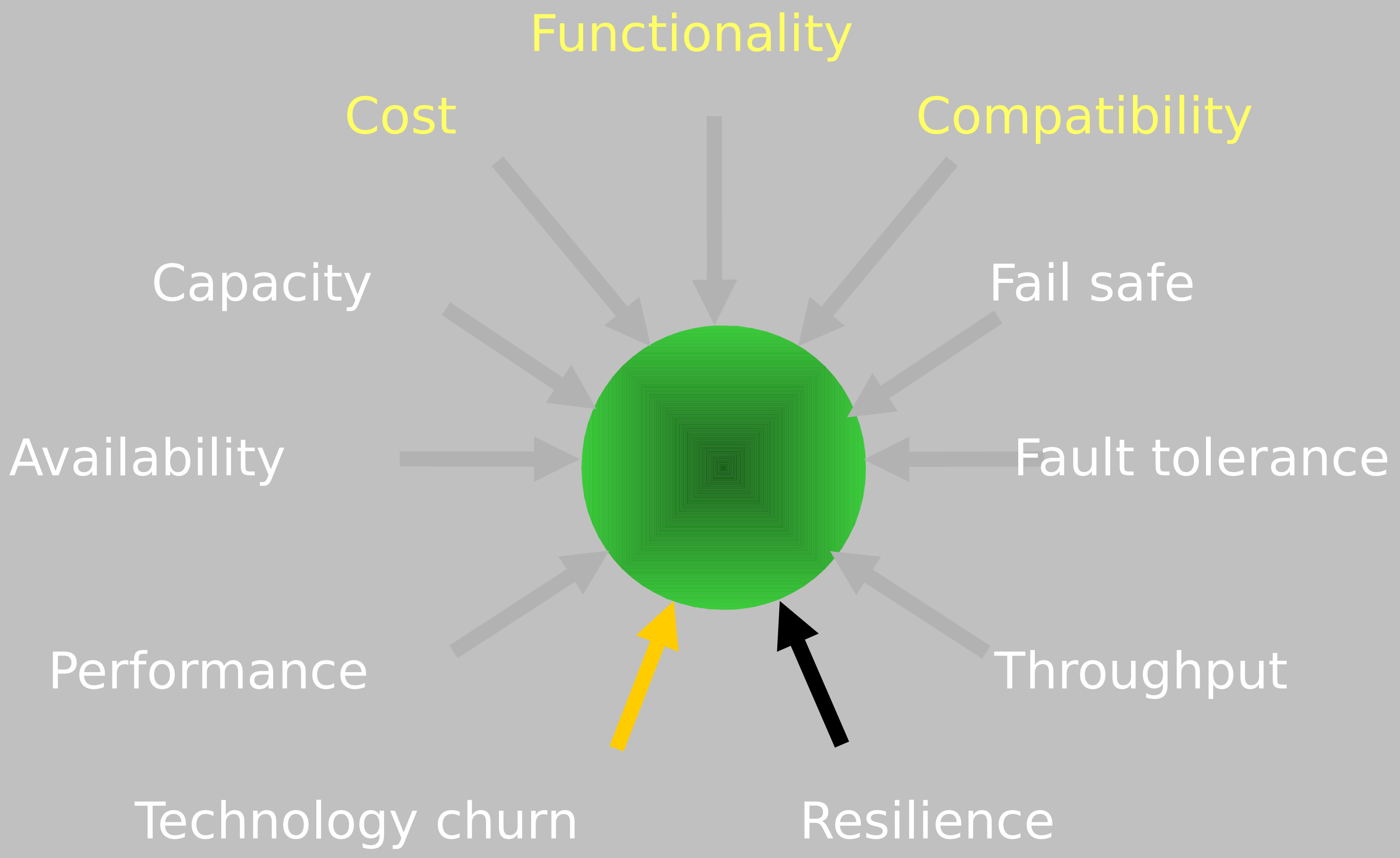
The domain of architecting



Architecture metamodel



Forces in Software



The challenge over the next 20 years will not be speed or cost or performance; it will be a question of complexity.
Bill Raduchel, Chief Strategy Officer, Sun Microsystems

Our enemy is complexity, and it’s our goal to kill it.
Jan Baan