



Describing Information Systems

Moving Beyond UML

Eoin Woods
Artechra
eoin@artechra.com

Nick Rozanski
Artechra
nick@artechra.com



Timetable

- 10:00 - 10:10 – Introductions
- 10:10 - 10:25 - **Presentation:** Architectural Description
- 10:25 - 10:40 - **Exercise 1:** "*Who, What, How ?*"
- 10:40 - 10:50 - Collect outputs of exercise
- 10:50 - 11:10 - **Presentation:** Architectural Notations
- 11:10 - 11:25 - **Exercise 2:** "*UML: Friend or Foe ?*"
- 11:25 - 11:30 - Collect outputs of exercise 2
- 11:30 - 11:45 - Break
- 11:45 - 12:25 - **Exercise 3:** "*Quivering at Arrows*"
- 12:25 - 12:45 - **Presentations** (5 minutes each)
- 12:45 - 13:00 - Summary and recap



Goals

- Existing description notations have proved to be weak in practice
- Could something better be designed?
- We'll try to do this during the session



Timetable

10:00 - 10:10 – Introductions

10:10 - 10:25 - **Presentation:** Architectural Description

10:25 - 10:40 - **Exercise 1:** "*Who, What, How ?*"

10:40 - 10:50 - Collect outputs of exercise

10:50 - 11:10 - **Presentation:** Architectural Notations

11:10 - 11:25 - **Exercise 2:** "*UML: Friend or Foe ?*"

11:25 - 11:30 - Collect outputs of exercise 2

11:30 - 11:45 - Break

11:45 - 12:25 - **Exercise 3:** "*Quivering at Arrows*"

12:25 - 12:45 - **Presentations** (5 minutes each)

12:45 - 13:00 - Summary and recap



What is Software Architecture

- The common definition:
 - *The software architecture of a program or computing system is the **structure or structures** of the system, which comprise software **elements** the externally visible **qualities** of those elements, and the **relationships** among them*
 - Len Bass, Paul Clements and Rick Kazman
Software Architecture in Practice, 2nd Edition



What is Software Architecture

- An alternative definition ...
 - *The set of system design decisions that dictate the fundamental structure and properties of a system*
 - *Thus, the set of decisions that will cause the system to fail if made incorrectly*
 - *The set of design decisions which, if made wrongly, cause your project to be cancelled!*

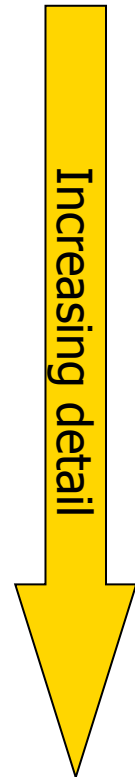


Stakeholders & Communication

- We build systems for stakeholders
- Architectural decisions are driven by stakeholder needs
- Effective communication with stakeholders is crucial
- Our architectural descriptions need to facilitate this
 - Our notations need to take stakeholders into account

Stakeholders & Communication

- Stakeholders differ in their needs
 - Acquirers
 - Assessors
 - Users
 - Trainers / technical writers
 - Administrators / technology experts
 - Developers & maintainers
 - Architects





Architectural Views

- Decompose an architectural description
- Target one or more stakeholder groups
- Focus attention on one piece of the problem
 - functional, deployment, information, ...
- Aid effective communication
 - appropriate representations for the view



Architectural Views



Functional View

Information View

Concurrency View

Development View

Deployment View

Operational View



Role of the Description

- Communicate the architecture
 - System overview (with selected detail)
 - Planning
- Ongoing reference
 - Architect, developers, testers
- Analysis of the architecture
 - Performance, availability, evolution, ...



Descriptive Difficulties

- The AD needs to serve different people
 - Different concerns, knowledge, assumptions, interests, amounts of time
- An AD contains *architectural* elements
 - Middleware, hardware, component types, connectors, information flows, ...
- The content required varies by context
 - Varying precision and detail



Exercise 1 – Who, What, How?

- Consider **who** needs to understand an architectural description
 - Consider the different views you might use
- Consider **what** they need to understand
 - Modules? Functions? Risks? Technologies?
- Consider **how** they use the description
 - Long term reference? Analyse the design? Input to further design? Quick overview of the system?



Exercise 1 – Who, What, How?

- Collect Outputs



Timetable

10:00 - 10:10 – Introductions

10:10 - 10:25 - **Presentation:** Architectural Description

10:25 - 10:40 - **Exercise 1:** "*Who, What, How ?*"

10:40 - 10:50 - Collect outputs of exercise

10:50 - 11:10 - Presentation: Architectural Notations

11:10 - 11:25 - **Exercise 2:** "*UML: Friend or Foe ?*"

11:25 - 11:30 - Collect outputs of exercise 2

11:30 - 11:45 - Break

11:45 - 12:25 - **Exercise 3:** "*Quivering at Arrows*"

12:25 - 12:45 - **Presentations** (5 minutes each)

12:45 - 13:00 - Summary and recap



Notations

- Architecture Description Languages
 - ACME, Wright, xADL, ...
- Boxes and Lines
 - Ad-hoc notation, usually in PowerPoint
- UML
 - De-facto standard but little real architecture notation



Notations - ADLs

- Many exist in the research domain
 - Wright, ACME, UniCon, ...
 - www.sei.cmu.edu/architecture/adl.html
- Few (none) have seen industrial use
 - Restrictive assumptions
 - Lack of multiple views
 - Lack of domain/technology specifics
 - Tools
 - Technology transfer



Notations - ADLs

A simple C/S System described in ACME (from CMU) ...

```

System simple_ca = {
  Component client = {
    Port send-request;
    Properties {
      Aesop-style : style-id = client-server;
      UniCon-style : style-id = ca;
      source-code : external = "CODE-LIB/client.c" }}

  Component server = {
    Port receive-request;
    Properties {
      idempotence : boolean = true;
      max-concurrent-clients : integer = 1;
      source-code : external = "CODE-LIB/server.c" }}

  Connector rpc = {
    Roles {caller, callee}
    Properties {
      asynchronous : boolean = true;
      max-roles : integer = 2;
      protocol : Wright = "..."}

  Attachments {
    client.send-request to rpc.caller ;
    server.receive-request to rpc.callee }
}

```

<http://www.cs.cmu.edu/~acme/>



Notations - ADLs

The XA three box model described in Rapide (from Stamford) ...

```

architecture LIA(NumRMS : Integer) for DTP is
  AP : Application Program(NumRMs);
  TM : Transaction Manager(NumRMs);
  RMs : array[Integer] of Resource Manager
      is (1 .. NumRMs, default is new(Resource Manager));
  connect
    AP.TX to TM.TX;
    for i : Integer in 1..NumRMs generate
      TM.XAs(i) to RMs[i].XA;
      RMs[i].AX to TM[i].AX;
      AP.AR(i) to RMs[i].AR;
    end generate;
end architecture LIA;

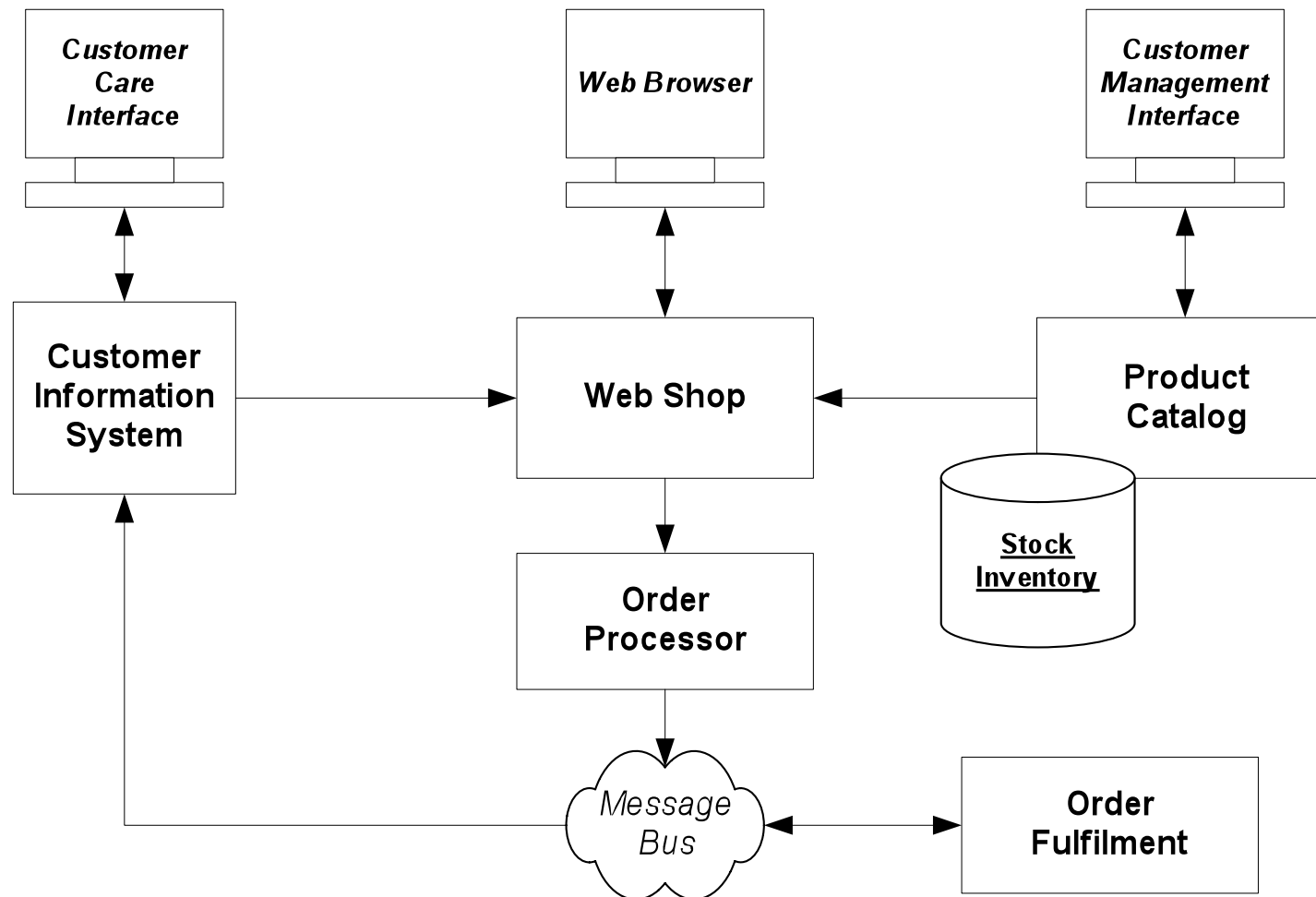
```



Notations - Boxes and Lines

- The most popular architectural notation
 - Flexible
 - Good tool support
 - Low learning curve
- Limitations
 - Ambiguity
 - Need to explain notation
 - Time to design notation

Notations - Boxes and Lines



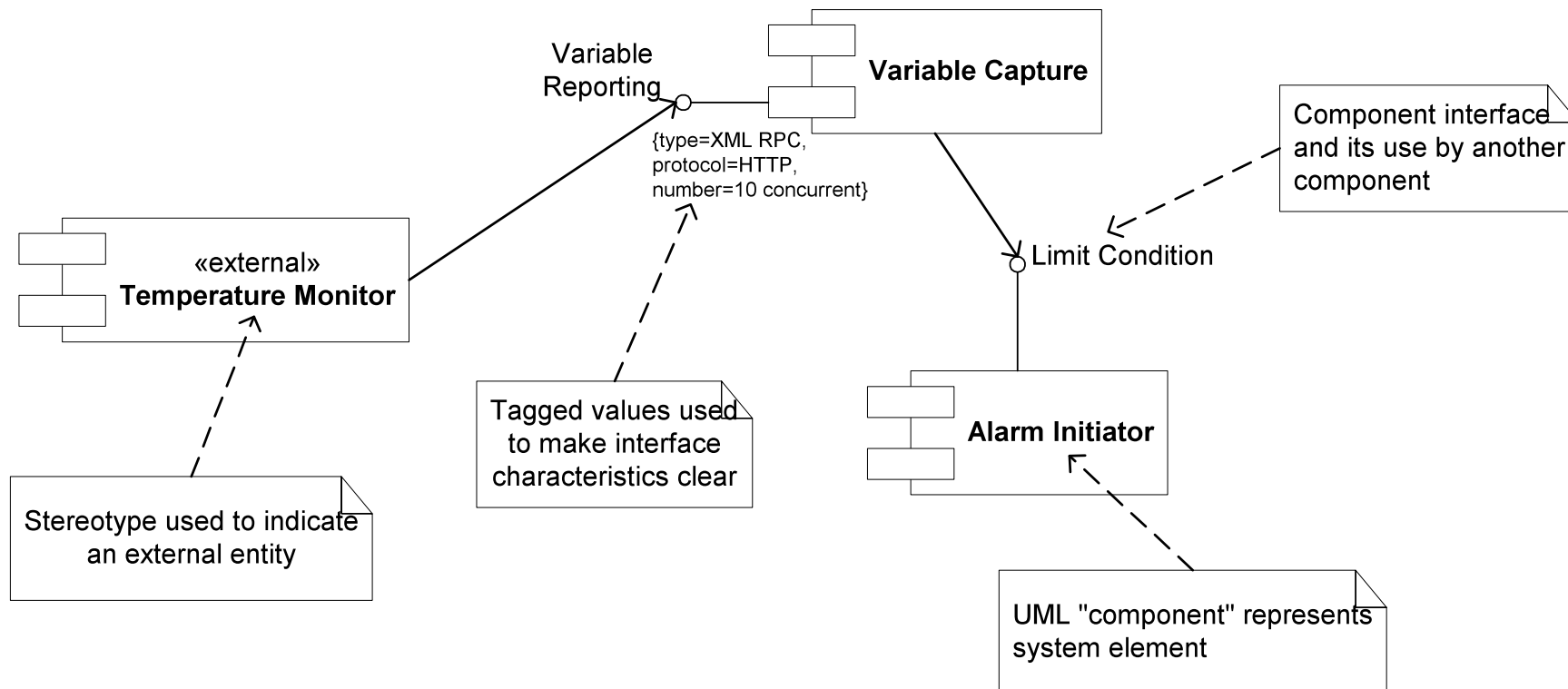


Notations - UML

- The de-facto “formal” notation
- General purpose software modeling language
 - Little specific architecture support
 - Needs abused or extended for architecture
- Widely understood, wide tool support
 - Although depth of understanding varies

Notations - UML

The UML component model ... one of UML's useful architectural models

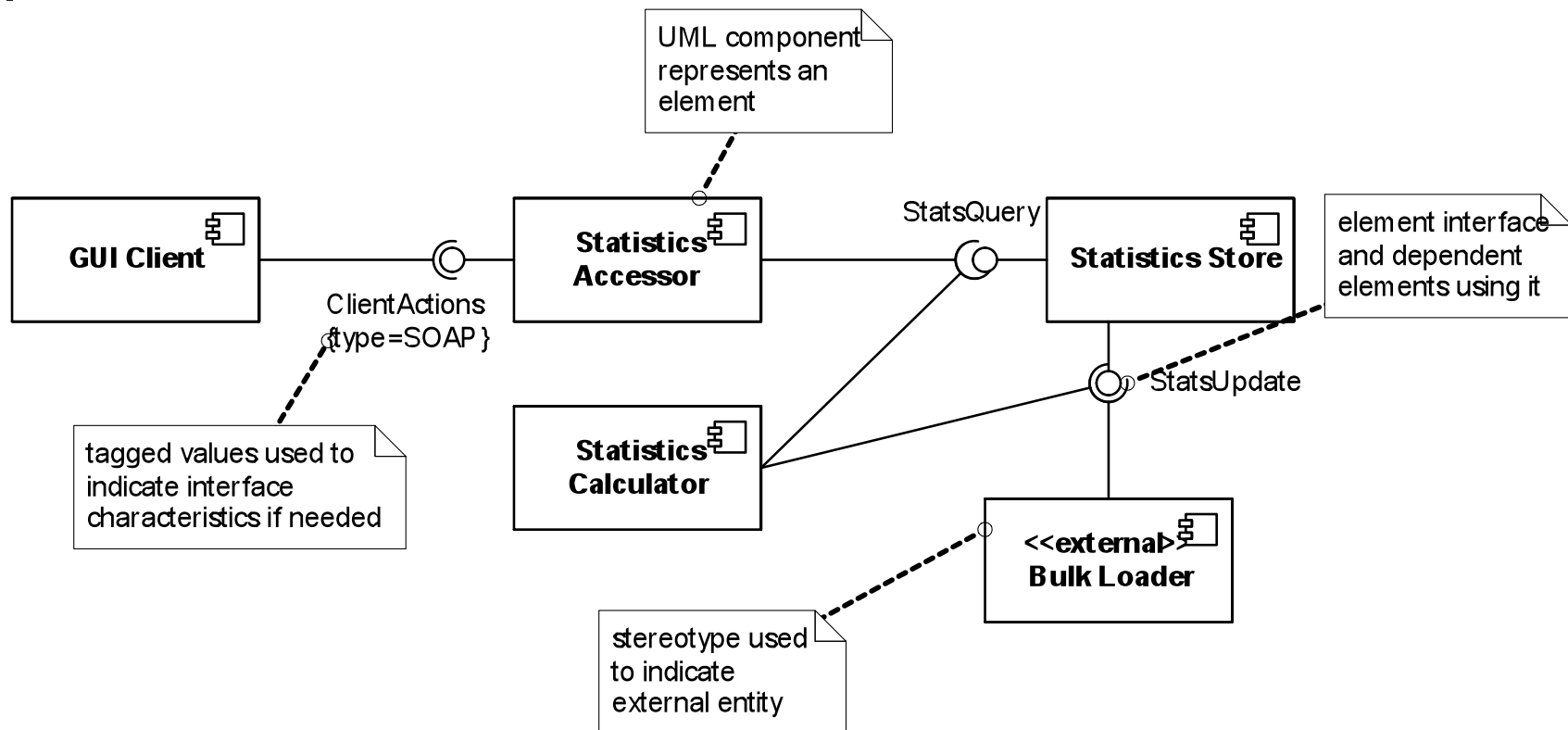




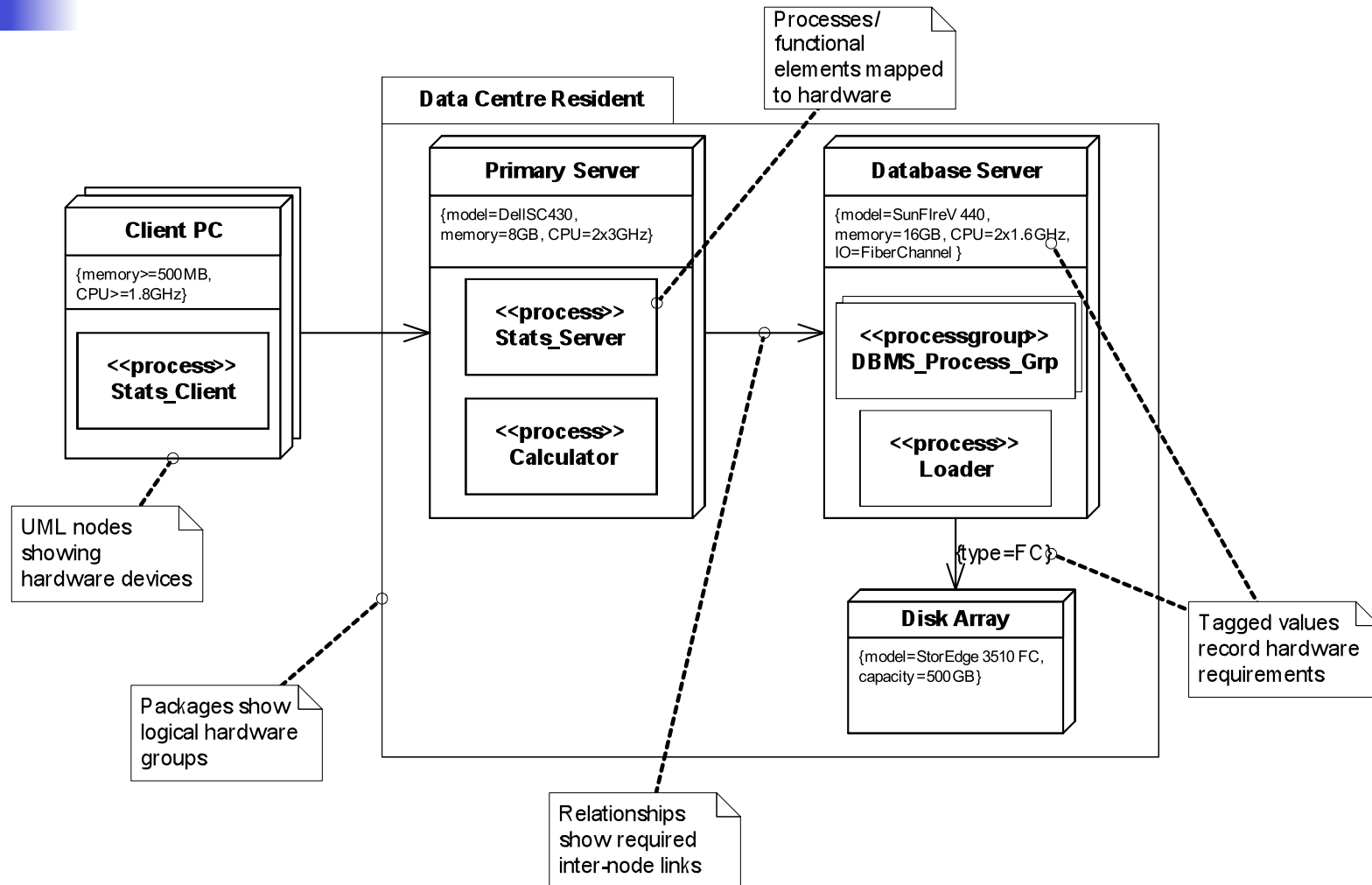
UML as an ADL

- UML is really an OOD notation
 - Grown over the years
 - Everything is a class
- Architectural constructs are basic
 - “Component”, interface, dependency
 - Node, link
- Architects lean heavily on extensions
 - Stereotypes, tagged values, notes(!)

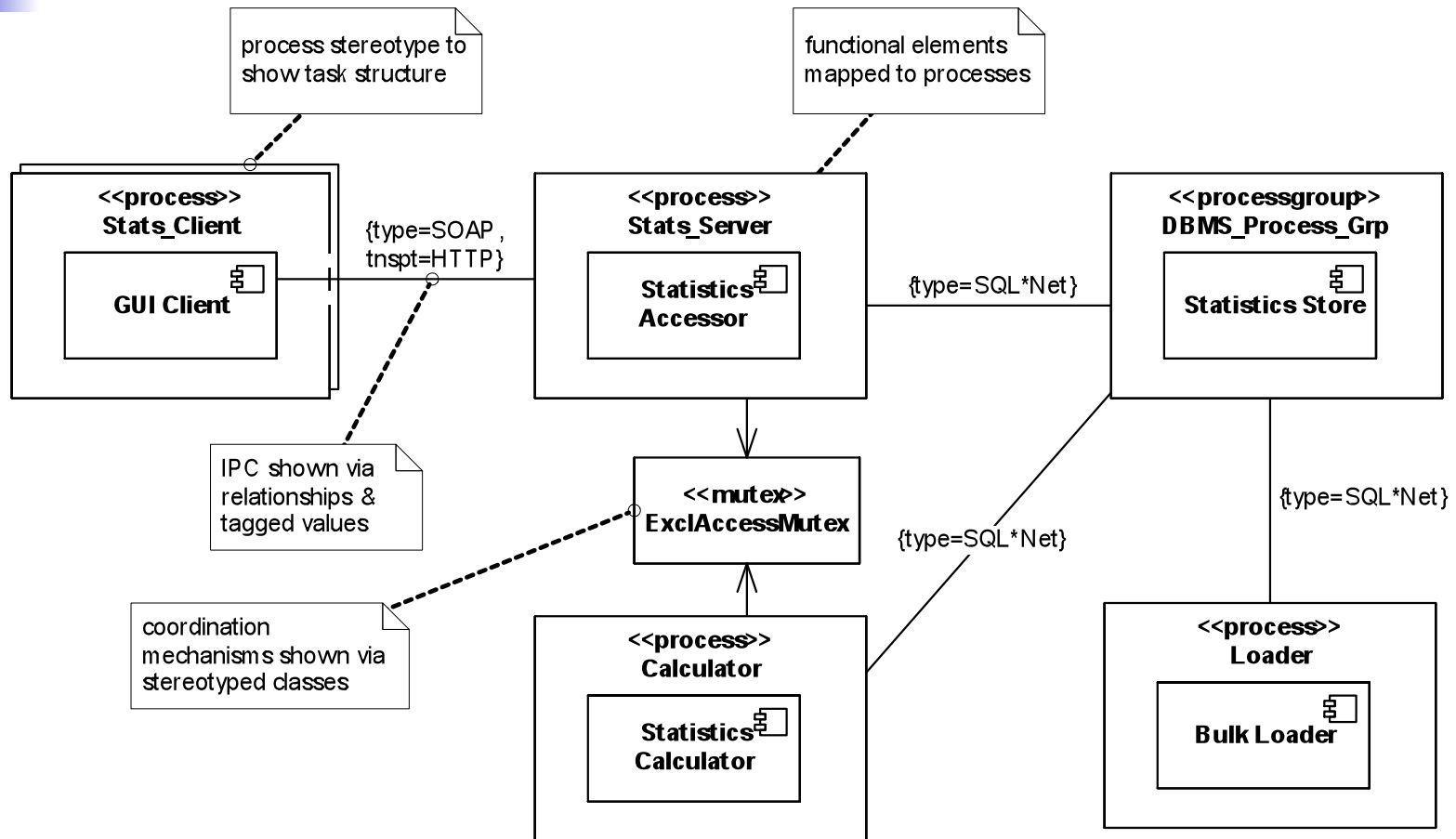
UML for Functional Structure



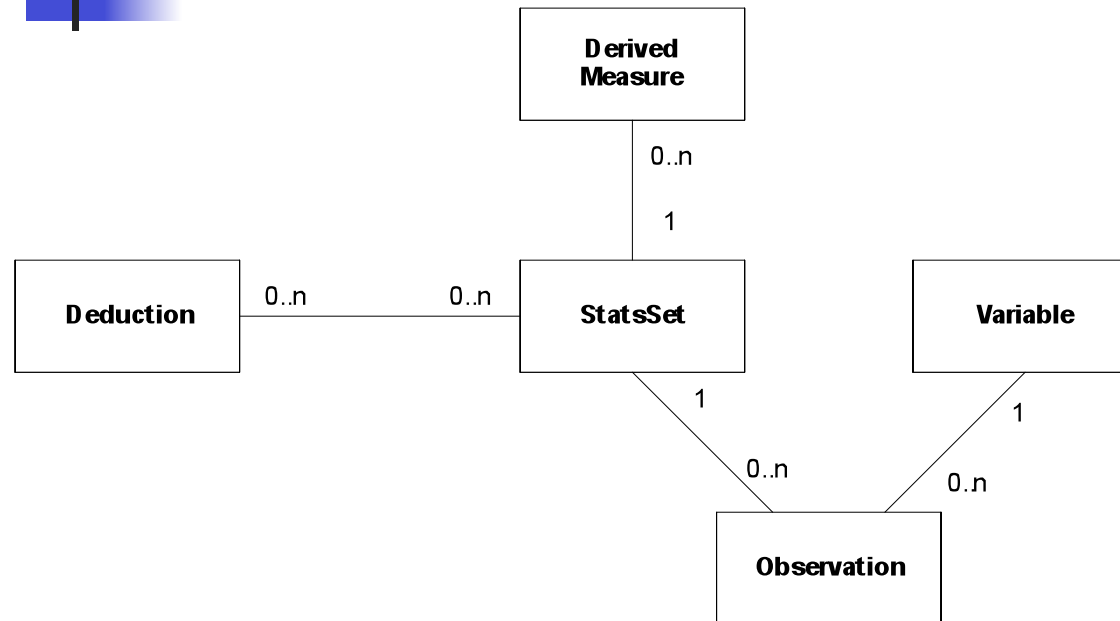
UML for Deployment Structure



UML for Concurrency Structure



UML for Information Structure



But how about

- Entity life history?
- Data flow?
- Volumetrics?
- Ownership?



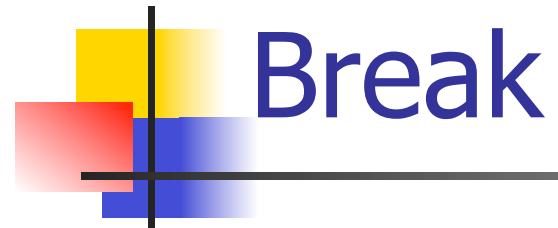
Exercise 2: UML, friend or foe?

- Establish how well standard UML meets the requirements identified in Exercise 1
- How effective would it would be for each of the stakeholder groups?



Exercise 2: UML, friend or foe?

- Collect Outputs



Back in 15 minutes please!

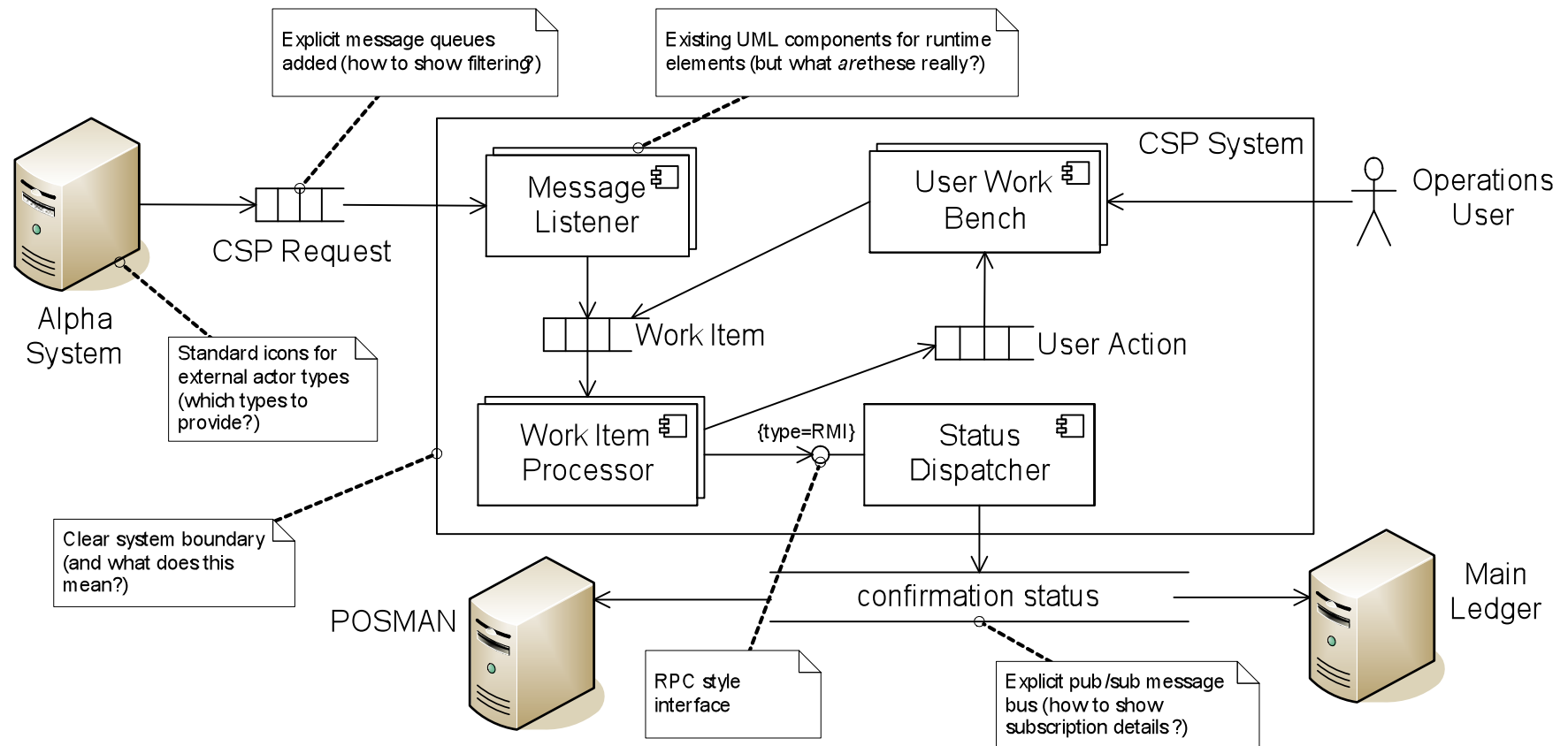


An Ideal ADL

- What would our ideal notation look like?
- What element types would it contain?
- What could it be used for?
- Whose needs would it address?
- What would make it different from existing approaches?

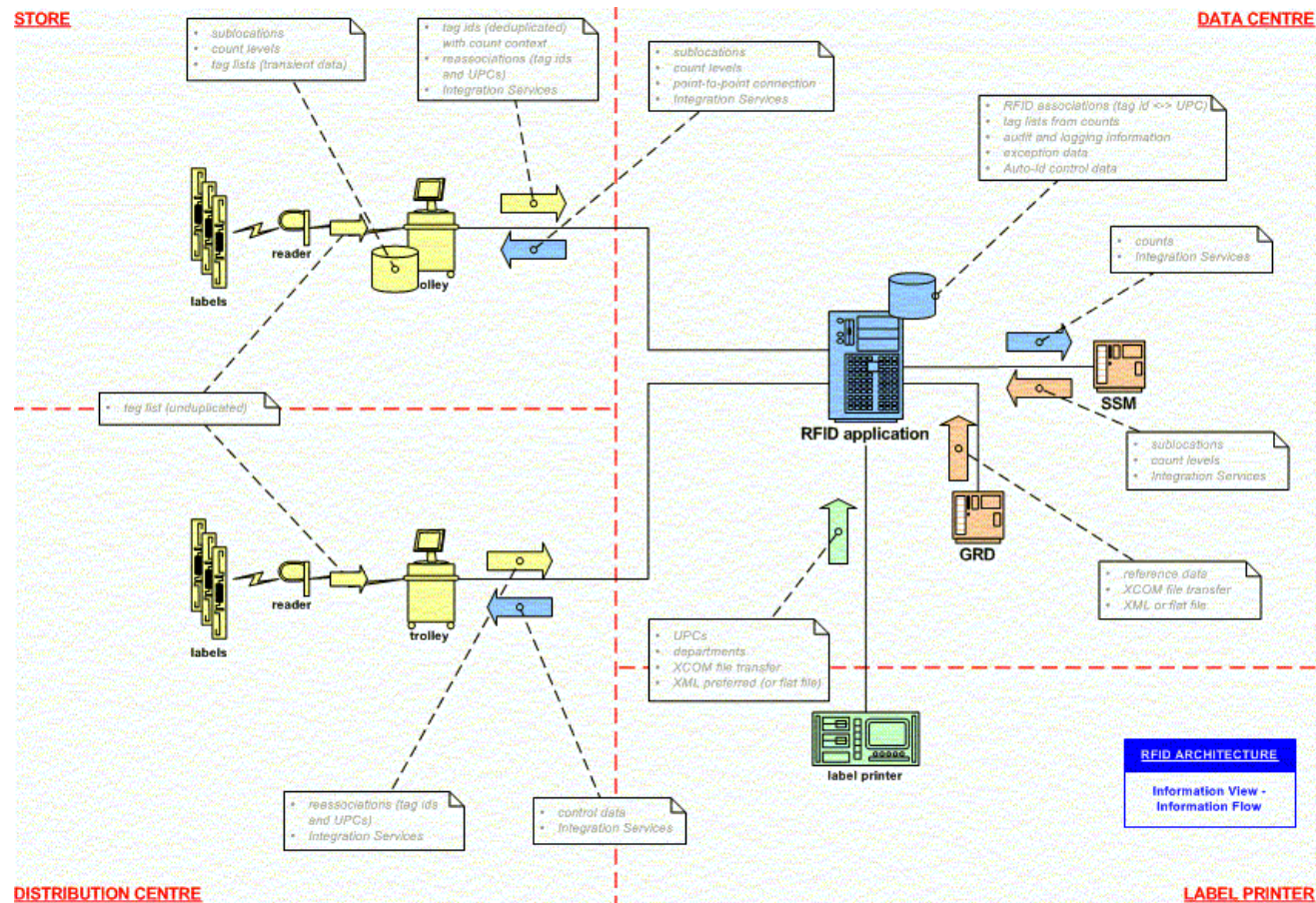
A Proto-ADL

One possibility ... a simple evolution and specialisation of UML



A Proto-ADL

Another example, for stakeholders who need a more informal and “pictorial” style





Exercise 3: Quivering at Arrows

- Attempt to improve on UML for architectural description
- Sketch a language for describing information systems
 - Language entities, relationships & semantics
 - Syntax (graphical and/or textual)
 - What it can be used for?
 - What tools would you need to provide?
 - Examples



Presentations

- Each group to present their language
- Keep presentations to about 5 minutes



Summary

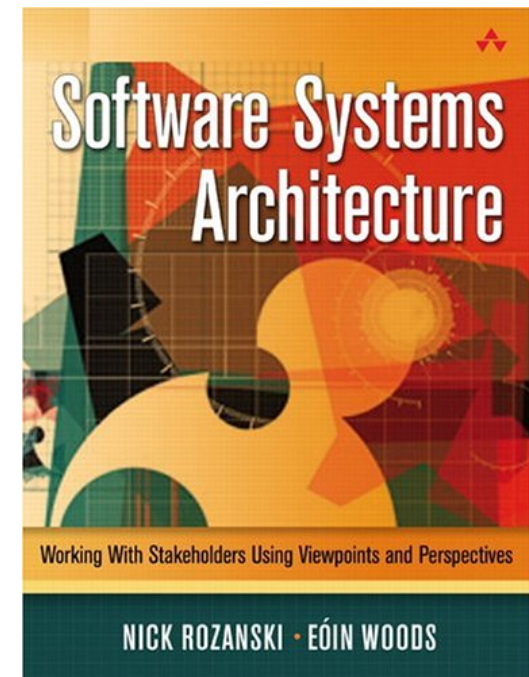
- Description is an important part of the architectural process
- Different stakeholders are looking for different things in a description
- Existing notations don't seem to have what we're looking for
- We've seen some ideas for new languages that might be more useful



More on Architectural Description

*Software Systems Architecture:
Working With Stakeholders
Using Viewpoints and
Perspectives*

Nick Rozanski & Eoin Woods
Addison Wesley, 2005



<http://www.viewpoints-and-perspectives.info>

Eoin Woods
Artechra
eoin@artechra.com

Nick Rozanski
Artechra
nick@artechra.com

Thank you