SOFTWARE ARCHITECTURE SUMMARY AND TAKEAWAYS

Suzanne Barber

Announcements

- Assignment #3
  - 2 DBs are required but don't have to be completely different
  - Decide what aspect of your DB1 you wish to change to create DB2 …. SB solutions? Style? SB solution infrastructure?
- Dec 9th Exam
  - Study, Study, Study the Study Guide
  - Plus today's lecture
- Dec 4th Presentations
  - Everyone in Class will evaluate presentations and submit reviews
  - Doors will close at 10:30am. Please be on time to attend !

And the Final SW Architecture Presentation winners are …

- **EE382C**
  - **Group 2**
- **EE397K**
  - **Group 13**
  - **Group 17**
  - **Group 12**
  - **Group 1-M**

Outline

- What was emphasized in this class (…and what to take away)
  - Theory and Practice
    - Definitions, motivations, representations, and process
  - Illustrated in the context of AWAREness methodology
    - BB, SB, and DB

SW Architecture Recap

- Topic Areas:
  - Definitions
    - What is an architecture
  - Motivations
    - Why bother
  - Representation
    - Notations
    - Underlying representations
  - Process
    - In the context of SW development process
    - Derivation
    - Evaluation

At some point in your career, you must be a SW Architecture "Evangelist"

- If someone asks you one of the following questions, would you have an educated answer?
  - What is a Software Architecture?
    - What's in it?
  - Why bother defining and maintaining a Software Architecture?
    - Who would use it and why would they care?
  - Can you describe the Software Architecture for system X?
    - You have five minutes with the CEO. How would you use it?
  - How do derive a software architecture?
    - Where does it come from?
    - Why did you structure it the way you did?

What *is* a SW architecture?

- Definitions – Two general categories
  - What a SW arch is composed of?
    - SW Arch = elements + form + rationale
    - Decomposition into parts and the communication between those parts
  - How a SW arch is used
    - Architectures serve as an important communication, reasoning, analysis and growth tool for systems
- Difference from other SW artifacts
  - Architecture prescribes/describes *structure*
  - Compare with graphical models used by Reqs. Engs.

Why bother with SW Architectures?

- Why bother
  - Support early and focused analysis
  - Manage change
  - Communicate among stakeholders (content stakeholder dependent)
  - Foster reuse
  - Manage complexity (abstraction)
  - Define and measure quality
  - Reduce development time
  - Analyze interfaces
  - …

SW Architecture Representations are critical for documentation and communication.

- Notation
  - Box-and-line diagram?
- Underlying representation
  - Architectural views separate concerns
    - AWAREness BB/SB/DB not the only representations
  - Importance of separation of concerns:
    - Emphasizes particular abstraction level
    - Easier to manage content
    - Focuses analysis
    - Captures particular stakeholder interests
    - Acknowledges that different requirements types evolve at different rates
      - Particularly what vs. how
      - Techs, sites, functionality

There is not one perfect notation/ representation for every need

SW Architecture Process – having disciplined process is more important than which one.

- SW development is a refinement/formalization process
  - Begin with informal requirements (natural language)
  - End with software – epitome of formal representation that (hopefully) embodies requirements
- Goals of a disciplined process
  - Manage complexity
  - Address different types of requirements
  - Deliver stakeholder demands on-time and within budget
- *Which* process doesn't matter

Some process is better than no process …

…and there's nothing wrong with customizing for a particular project/organization.

The most important feature of a disciplined architecture derivation method is the capture of *rationale and traceability*.

There is a (continuous) spectrum of architectures capturing (1) Different abstraction levels ad (2) Different concerns

- The functionality-to-appl-to-site process not set in stone
- So…a disciplined process could derive/evaluate architectures in *any* order
  - E.g., site resources are fixed and particular technologies have already been chosen

Disciplined Architecture Derivation requires Disciplined Decision Making.

- General decision making process
  - What's the problem
  - Gather information
  - Identify alternatives
  - Make a selection
  - Gather feedback from your decision
- General architecture derivation/evaluation process (for each view -- BB, SB, and DB)
  - Determine type of architecture needed
  - Determine requirements to be met
  - Identify potential components and connectors
    - often inferred from what you put into the components
  - Select components and connectors
  - Evaluate

There is no one perfect architecture BUT …

- "If it is true that, given the same technical requirements for a system, two different architects in different organizations will produce different architectures, how can we determine if either one of them is the right one?" [Bass et.al., 1998]
  - Business Blueprint judged by which criteria?
  - Solution Blueprint judged by which criteria?
  - Deployment Blueprint judged by which criteria?

What are some key challenges a disciplined architecture derivation and evaluation process must address?

- Many and sometimes competing decisions
  - tradeoffs and conflicts
- Justifying Rationale and traceability
- Lots of stakeholders involved
  - Stakeholders using and influencing the architecture
- Need one decision-maker – Architect, capable of hearing stakeholders, making reasonable and rationalized trade-offs, communicating and documenting decisions with traceability and support.
- Change demanding responsive iteration
- Planning for the short-term and horizon.

Architectural styles and patterns offer guidance to the architect.

- Styles and patterns offer …
  - Bottom up, Proven form, experience
  - Influence any architecture level
  - Hybrids
  - Relate to qualities

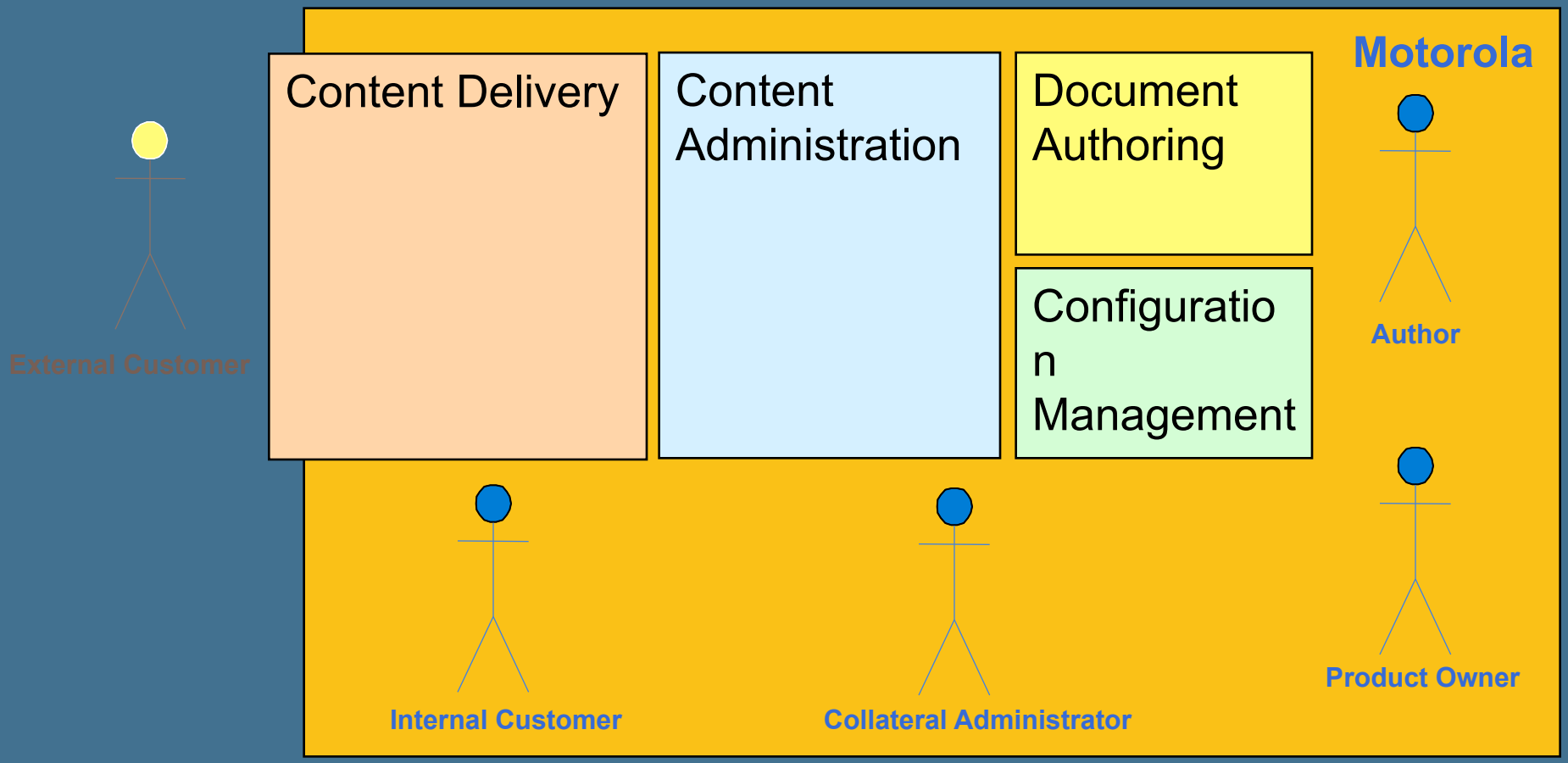
Architectures must be measured to evaluate progress towards goals and determine next steps.

- What are your concerns?
- How to measure and compare?
  - Stakeholder verification included
- What representation necessary to accommodate?
- What evaluation technique to accommodate?
- Architect must ..
  - Link high-level fuzzy qualities to low-level metrics

New Material regarding Architectural Evaluation

Focus on the definitions of measurements for safety, liveness, and reliability.

eDesign Business Blueprint



Some eDesign Usage Profiles

Usage Profile	*Frequency*	*Tasks*	*Performer*	*Duration*
Publish New Collateral	medium	Add Collateral File	Collateral Admin	high
		Launch Collateral	Collateral Admin	low
Publish New Product	medium	Create Product Summary Page	Product Owner	high
		Associate Collateral to PSP	Product Owner	med
		Launch PSP	Product Owner	low
		Associate PSP to a Category	Product Owner	low
Access Product Information	high	Search for Product	Customer	med
		Select PSP	Customer	low
		Download Product Collateral	Customer	high
Publish New Technical Document	medium	Create Technical Documentation	Author	high
		Create Technical Document Metadata	Author	high
		Check In Technical Documentation	Author	low
		Check In Technical Document Metadata	Author	low
		Stage Technical Documentation	Publisher	med
Update Collateral	low	Stage Collateral	Collateral Admin	low
		Update Collateral Attributes	Collateral Admin	high
		Launch Collateral	Collateral Admin	low
Update Product	low	Stage Product	Product Owner	low
		Update Product	Product Owner	high
		Launch Product	Product Owner	low

Arcade Safety Property

Informally - “the system does not terminate in an invalid end state”, including:

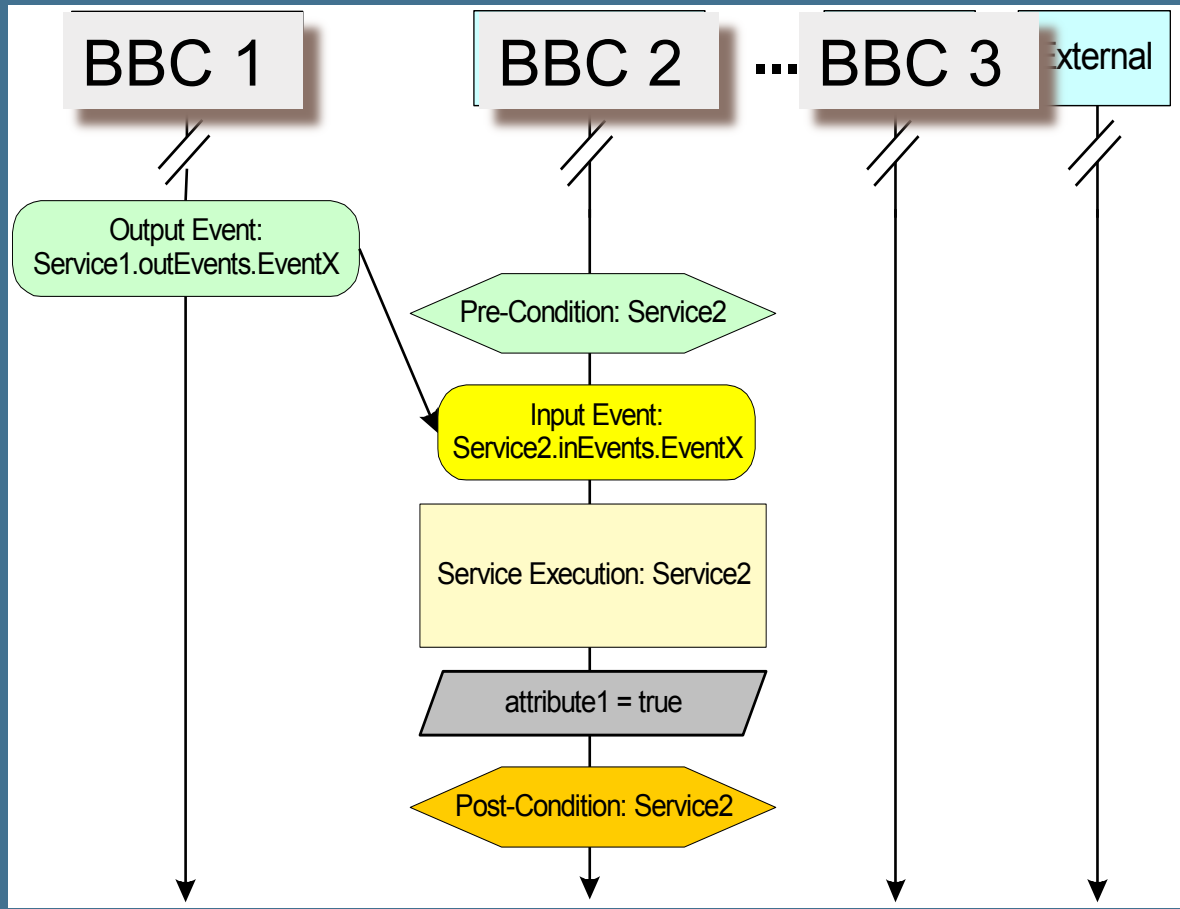
- Unterminated service executions
- Unreceived events or data

Unterminated service executions occur when post-conditions cannot be satisfied.

Unreceived events or data occur as a result of pre-condition or post-condition specification errors.

Arcade presents SPIN safety property violation counterexamples using an Architecture Trace Diagram (ATD), an extension to ITU Message Sequence Charts

Architecture Trace Diagrams for each Scenario to measure Safety



LEGEND

Pre-condition satisfaction:
green hexagon, annotated with name of the associated service

Post-condition satisfaction:
orange hexagon, annotated with name of the associated service

Service execution:
tan rectangle,annotated with name of the service

Sending of a datum or event:
green rounded rectangle, annotated with name of the associated service and name of datum or event sent

Reception of a datum or event:
yellow rounded rectangle, annotated with name of the associated service and name of datum or event received

Assignment of an attribute:
gray parallelogram, annotated with name of the associated attribute

Arcade Liveness Property

Informally – "the system eventually enters all desirable states", including:

- **No unreachable services exist**
- **All required paths between services are traversable**

Unreachable services occur when an entire service pre-condition is never satisfiable.

Untraversable paths occur when a disjunct sub-expression of a service pre-condition is never satisfiable.

Arcade presents SPIN liveness property violation counterexamples by indicating which pre-conditions are not satisfiable.

eDesign Correctness Evaluation Results

57% of errors were specification errors

43% of errors were conceptual integrity errors

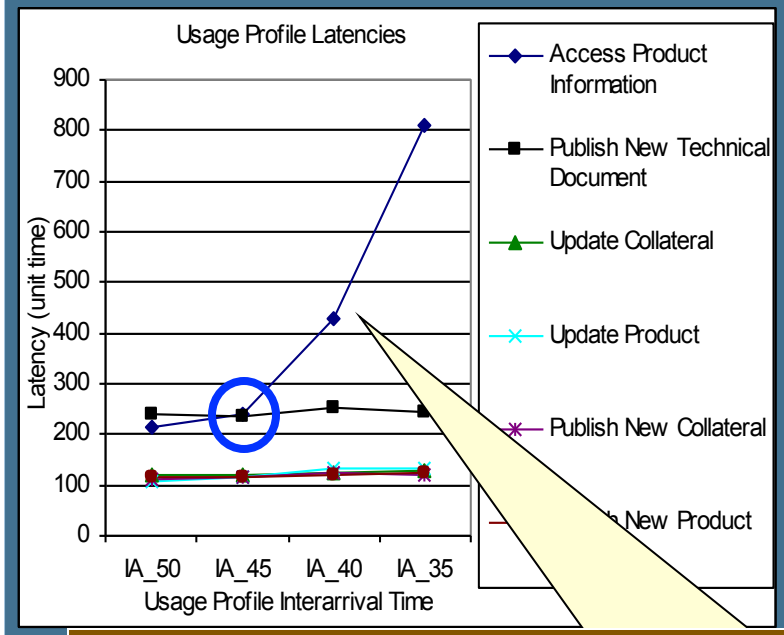
BBC revision	Safety Errors	Liveness Errors	Completeness Errors	Detected by Simulation	Detected by Model Checking
1	1	0	6	6	1
2	10	4	4	4	14
3	6	3	1	1	9
4	3	1	1	1	4
5	0	0	0	0	0
totals	20	8	12	12	28

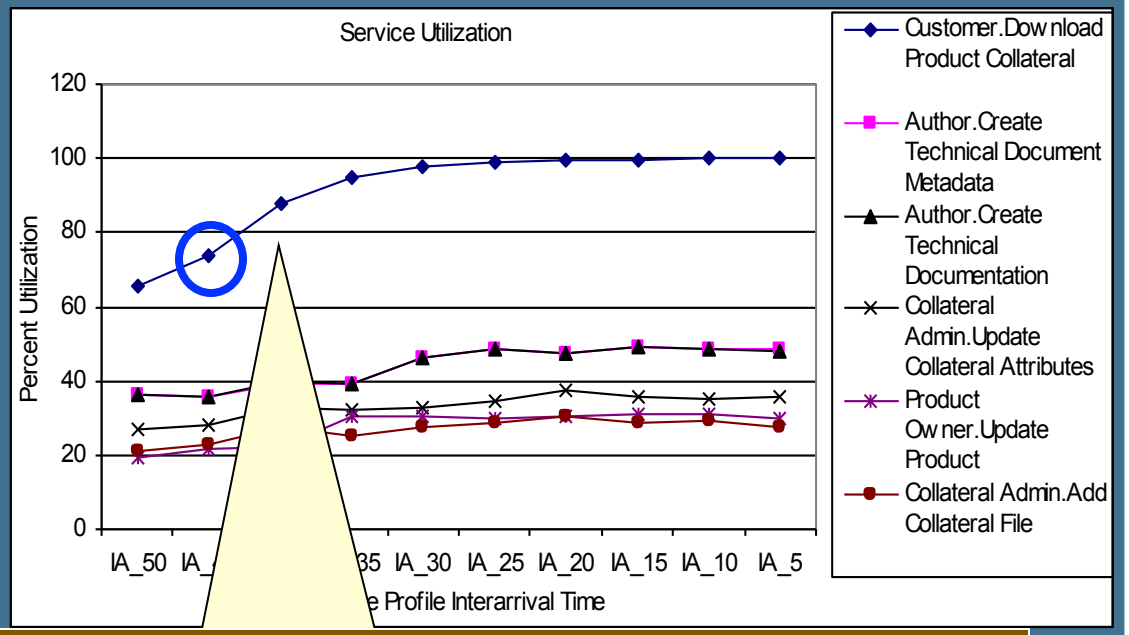
Arcade Reliability Evaluation

Arcade evaluates how sensitive the reliability of an Architecture is to the reliability of individual elements of the architecture.

Goal is to help stakeholders understand the overall system reliability as a function of reliability of services, Business Blueprint Components, etc.

eDesign Performance Evaluation Results for Implementation Guidance





Implementation Guidance:

System Implementers should seek to design the system so that the utilization of "Download Product Collateral" does not exceed 75%

Collateral" Service is part of the "Access Product Information" Usage Profile.

Evaluation: Different requirements and different questions demand different representations and evaluation methods

- Can be with respect to:
  - Components
  - Connectors
  - System

You cannot validate that your architecture meets stakeholder objectives if you cannot *measure* (in some manner).

Decide *what to evaluate*,

what evaluation type is appropriate,

what representation will enable that evaluation, and

how you will compare and communicate the results.