Documenting the Architecture

Some of the material in these slides is taken from Software Architecture in Practice, 2nd edition by Bass, Clements and Kazman.
The ADL slides are based on a presentation by T. Cook of Microelectronics and Computer Technology Corporation

Different Needs of Stakeholders

- Different stakeholder groups have different needs
- Should provide different views to satisfy those needs
- Often create one document with different roadmaps for different groups

Stakeholders and Views

	Module Views				C&C Views	Allocation Views	
Stakeholder	Decomposition	Uses	Class	Layer	Various	Deployment	Implementation
Project Manager	s	s		s		d	
Member of Development Team	d	d	d	d	d	s	s
Testers and Integrators		d	d		s	s	s
Maintainers	d	d	d	d	d	s	s
Product Line Application Builder		d	s	o	s	s	s
Customer					s	o	
End User					s	s	
Analyst	d	d	s	d	s	d	
Infrastructure Support	s	s		s		s	d
New Stakeholder	x	x	x	x	x	x	x
Current and Future Architect	d	d	d	d	d	d	s

Key: d = detailed information, s = some details, o = overview information, x = anything

Choosing Relevant views

- Produce a candidate view list
- Combine views
- Prioritize

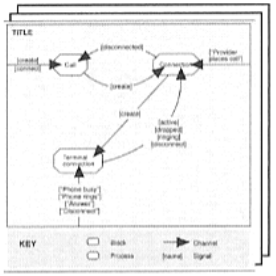
Suggested View Documentation Template

1. Primary presentation – elements and their relationships
2. Element catalog -- explains the picture
3. Context diagram -- how the system relates to its environment
4. Variability guide – how to exercise any variation points
5. Architecture background – why the design reflected in the view came to be
6. Glossary of terms used
7. Other information

Pictorially

Views

Section 1. Primary Presentation of the View



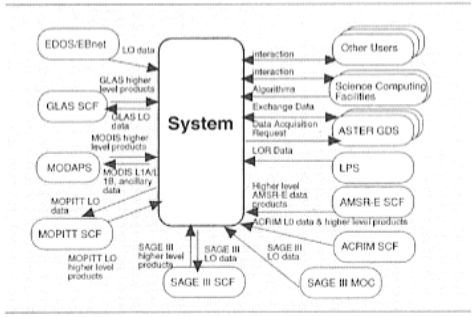
OR

Textual version of the primary presentation

Section 2. Element Catalog

- Section 2.A Elements and their properties
- Section 2.B Relations and their properties
- Section 2.C Element interfaces
- Section 2.D Element behavior

Section 3. Context Diagram



Section 4. Variability Guide

Section 5. Architecture Background

- Section 5.A Design rationale
- Section 5.B Analysis of results
- Section 5.C Assumptions

Section 6. Glossary of Terms

Section 7. Other Information

Documenting Interfaces

1. Interface identity - unique name
2. Resources provided
3. Locally defined data types - if used
4. Exception definitions - including handling
5. Variability provided - for product lines
6. Quality attribute characteristics - what is provided?
7. Element requirements
8. Rationale and design issues - why these choices
9. Usage guide - protocols

Architecture Description Languages

ADL -definition

"form of expression used for the description of architectures"

- ISO/IEC 42010

ADLs - Positives

- Provide a (formal) way of representing architecture
- Intended to be human and machine readable
- Support describing a system at a higher level than previously possible
- Permit analysis of architectures – completeness, consistency, ambiguity, and performance
- Can support automatic generation of software systems

ADLs - Negatives

- No universal agreement on what ADLs should represent, particularly as regards the behavior of the architecture
- Representations currently in use are relatively difficult to parse and are not supported by commercial tools
- Most ADL work today has been undertaken with academic rather than commercial goals in mind
- Most ADLs tend to be very vertically optimized toward a particular kind of analysis

Candidate ADLs

- Leading candidates
  - ACME (CMU/USC)
  - Rapide (Stanford)
  - Wright (CMU)
  - Unicon (CMU)
- Secondary candidates
  - Aesop (CMU)
  - MetaH (Honeywell)
  - C2 SADL (UCI)
  - SADL (SRI)