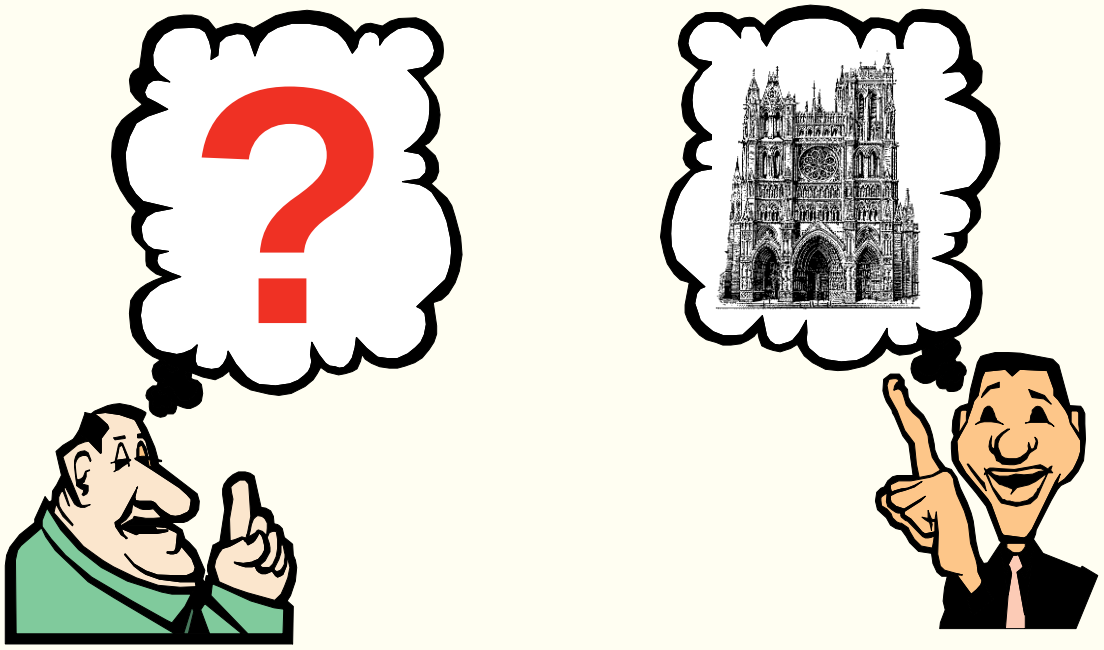
What is An Architecture?

Kenneth J. Turner

Department of Computing Science and Mathematics
University of Stirling, Stirling FK9 4LA, Scotland

Telephone: +44-1786-467-420 *Facsimile:* +44-1786-464-551
Email: kjt@cs.stir.ac.uk *Web:* http://www.cs.stir.ac.uk/~kjt/

7th September 1999



The Importance of Architecture

- *architecture* not *development* is hard
- architecture like program design, development like program coding
- poor architecture means:
  - hard to understand
  - hard to modify and extend
  - hard to analyse
- a good architecture:
  - essential for vague, ill-defined areas
  - improves informal problem statement
- working on a messy problem directly leads to a messy design!

Some Definitions

- dictionary definitions of architecture:
  - art or practice of structures
  - unifying or coherent form
  - method or style of building
- dictionary definitions of formality:
  - form or essence of a thing
  - outward form, structure, relationships or arrangement rather than content
- dictionary definitions of specification:
  - arrangement in a definite pattern
  - organisation of parts dominated by the whole
  - aggregated elements and their relationships
- architecture, formality and specification are thus closely related!

Historical Views

- ‘in architecture, as in all the other operative arts, the end must direct the operation’ *(Sir Henry Wotton, 1624)*
- ‘the four arts of poetry, painting, music and architecture (which is a science) are the four faces of man’ *(William Blake, 1808)*
- ‘architecture in general is frozen music’ *(Friedrich von Schelling, 1809)*
- ‘no person who is not a great sculptor or painter can be an architect, he can only be a builder’ *(John Ruskin, 1853)*
- ‘no architecture is so haughty as that which is simple’ *(John Ruskin, 1853)*
- ‘after great pain, a formal feeling comes’ *(Emily Dickinson, 1876)*

Questions and Answers

- questions – the nature of architecture:
  - is architecture merely structure?
  - is architecture always design?
  - is structure mandatory?
  - do style and elegance play a part?
  - what should an architecture contain?
  - how detailed should architecture be?
  - how to assess architecture quality?
  - how to handle legacy architectures?
- answers – architecture concerns:
  - components (the building blocks)
  - combinators (how to combine the building blocks)
  - principles (guidelines for defining the architecture)
  - criteria (methods for assessing the architecture)

Architectural Principles

- architectural principles answer questions like:
  - how to get a good architecture?
  - is this a good architecture?
  - is this architecture better?
- such principles are only *guidelines* and not mechanistic
- architectural principles come from:
  - systems theory
  - software engineering
  - formal methods

Principle	**Techniques**	**Criteria**
Modularity	Functional Decomp. Constraint Decomp. Temporal Decomp. Spatial Decomp.	Coherence Decoupling Proportion
Generality	Parameterisation Generalisation Unification	Abstractness Commonality Adaptability
Simplicity	Idealisation Deferment Minimisation	Uniformity Elegance Economy

Modularity

- some familiar techniques:
  - divide and conquer
  - structured analysis
  - top-down specification
- modularity techniques:
  - functional decomposition
  - constraint decomposition
  - temporal decomposition
  - spatial decomposition
- decomposition carried out ‘vertically’, so multiple hierarchical levels

Generality

- must strike a balance between:
  - specificity – suitable for next refinement, but not too specialised
  - generality – more general than needed *now*, but not too cumbersome
- generality techniques:
  - parameterisation – values, structure (e.g. for replication), sensible defaults
  - generalisation – avoiding *unnecessary* restrictions
  - unification – identifying similarities

Simplicity

- simplicity works with generality, removing *unnecessary* differences
- may require greater abstractness though not generality
- simplicity techniques:
  - idealisation:
    - ‘blue sky’ approach
    - initially ignore restrictions
    - avoid special cases
  - deferment:
    - avoid details too soon
    - controlled introduction of structure
  - minimisation
    - one solution instead of two (‘Occam’s Razor’)
    - consistent use of limited constructs

Personal Publications on Architecture

- Gyula Csopaki and Kenneth J. Turner. Modelling digital logic in SDL. FORTE X/PSTV XVII, November 1997.
- Richard O. Sinnott and Kenneth J. Turner. Applying the architectural semantics of ODP to develop a trader specification. Computer Networks, March 1997.
- Kenneth J. Turner. An architectural foundation for relating features. Feature Interactions in Telecommunication Networks and Software Systems, June 1997.
- Kenneth J. Turner. Incremental requirements specification with LOTOS. Requirements Engineering, November 1997.
- Kenneth J. Turner. Relating architecture and specification. Computer Networks, March 1997.
- Kenneth J. Turner. Specification architecture illustrated in a communications context. Computer Networks, March 1997.
- Kenneth J. Turner. A structural comparison of FrameMaker and LaTeX. Electronic Publishing, April 1997.
- Kenneth J. Turner. An architectural description of intelligent network features and their interactions. Computer Networks, September 1998.
- Kenneth J. Turner and Richard O. Sinnott. DILL: Specifying digital logic in LOTOS. FORTE VI, October 1994.
- Kenneth J. Turner and Marten van Sinderen. LOTOS specification style for OSI. The LOTOSPHERE Project, 1995.