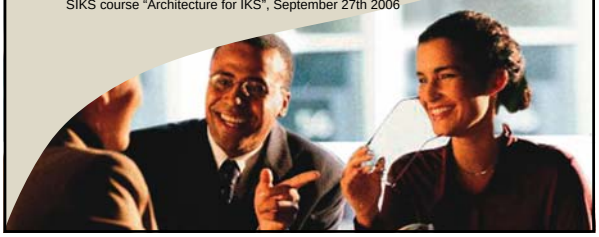
Need for architecture

Maarten Waage, enterprise architect (Capgemini)
SIKS course "Architecture for IKS", September 27th 2006



Topics addressed

- Definitions of architecture
- Need for architecture
- Types of architecture
- Stakeholders and their concerns
- Architecture principles
- Views and frameworks
- Role of the architect

Architecture definition: IEEE 1471-2000

- "An architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution."

Source: IEEE STD 1471-2000

Architectural description uses IEEE 1471-2000

- Architectural descriptions are applicable to a variety of uses, by a variety of stakeholders, throughout the life cycle. These uses include, but are not limited to the following:
  - Analysis of alternative architectures
  - Business planning for transition from a legacy architecture to a new architecture
  - Communications among organizations involved in the development, production, fielding, operation, and maintenance of a system
  - Communications between acquirers and developers as a part of contract negotiations
  - Criteria for certifying conformance of implementations to the architecture
  - Development and maintenance documentation, including material for reuse repositories and training
  - materials
  - Input to subsequent system design and development activities
  - Input to system generation and analysis tools
  - Operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components
  - Planning and budget support
  - Preparation of acquisition documents (e.g., requests for proposal and statements of work)
  - Review, analysis, and evaluation of the system across the life cycle
  - Specification for a group of systems sharing a common set of features, (e.g., product lines)

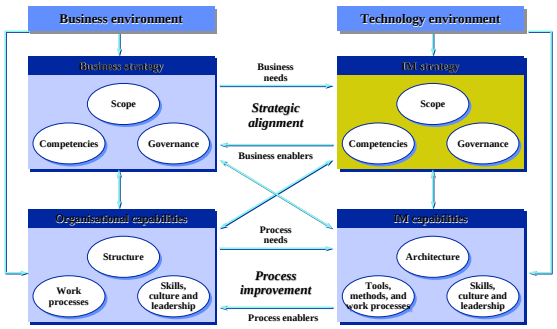
Source: IEEE Std 1471-2000

Architecture definition: TOGAF

- The Open Group's Architectural Framework (TOGAF)
- In TOGAF, "Architecture" has two meanings depending upon its contextual usage:
  - A formal description of a system, or a detailed plan of the system at component level to guide its implementation.
  - The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.

Source: The Open Group (www.opengroup.org/architecture)

Model for business strategy - IT strategy alignment



from N. Venkatraman and J.C. Henderson in "IBM Systems Journal"

System development

- Aim:
  - Produce systems that conform to the desires
  - Including desires wrt the development process
- Challenge:
  - Reduction of uncertainty wrt the desiredness of:
    - a (future) system's behaviour
    - behaviour of the development process

Uncertainties

- Who / what determines desiredness?
- Agreement / commitment to specifications?
- Completeness of requirements?
- Completeness of designs
- Does the system / process match the specifications?
- …

Trade-off

- Potential costs due to undesired behaviour
  vs
  Potential costs of reducing uncertainty
- Challenge:
  - Requires insight into the issues involved
  - Requires decisions from stakeholders and designers
- Means:
  - Architecture!?

Role of architecture

- Some goal / need to state common properties on the design of a class of systems:
  - Selecting among alternatives
  - Defining core characteristics of a family
  - Guiding evolution of incarnations
- Relative to some higher goal
  - Cost effectiveness?
  - Creating the architecture, maintaining and enforcing it
  - Versus the benefit of insight & control

Role of architecture

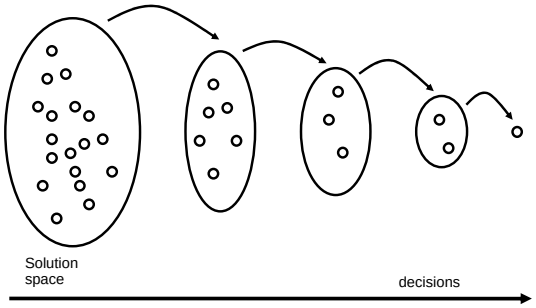
- A need to state, discuss, negotiate and communicate common properties:
  - Selecting among alternatives
  - Defining core characteristics of a family
  - Guiding evolution of incarnations
- Requirements on architecture as a means:
  - Provide relevant insight
  - Communicatable
  - SMART enough to govern / steer
  - ….

Architecture definition

- "In general, the design freedom of designers is undesirable large. The idea of architecture is to take advantage of this. Therefore, architecture is defined as normative restriction of design freedom.
  This idea of consciously applying normative restriction of design freedom is the really new thing. It makes architecture a prescriptive notion; any descriptive interpretation is cogently rejected"

Source: xAF/NAF

Definition & designing ...



Defining & designing ...

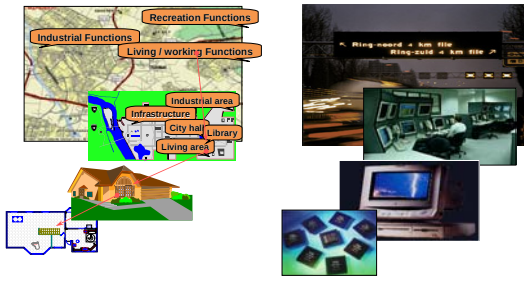
- Each decision limits the solution space
- A restriction of design freedom!

- What makes architecture so special?
- There must be an explicit goal to
  - state, discuss, communicate or negotiate common properties of a
  - family, incarnations or equivalence class of systems

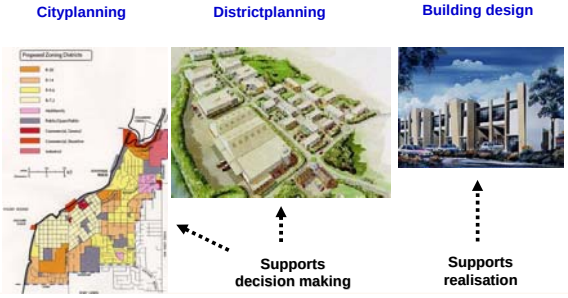
Limiting design freedom

- Can be done by prescriptive
  - (partial) “models” / “reference models”
  - design principles

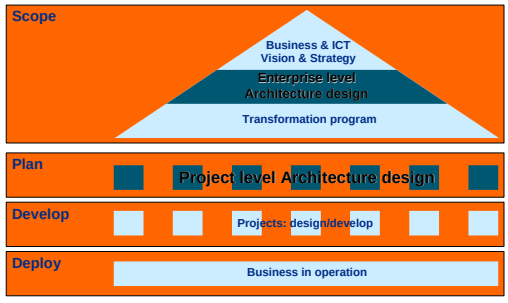
An architecture design may have varying levels of granularity



Enterprise architecture versus solution architecture



Different architecture levels



Architecture ALWAYS delivers a solution to a business problem

Business issue	Example	Architecture Objective
Transformation	E-business transformation	Create structure to build and support: •Business case •Migration plan
Rationalisation	Clean up the mess after acquisitions & mergers	Focus on defining standards etc to guide rationalization
Integration	Implement CRM to interface to 200 back office systems	Create (detailed) structure to be able to understand all the interfaces etc.
Optimisation	Reduce development Time-to-market	Raise the quality of the development process
Business – IT alignment	Link process development and its IT support	Provide better IT support

Goals of architecture

- Atlas for management
  - For positioning decisions and making impact-of-change transparent
  - To increase alignment (business-ICT, strategic-operational)
- Instrument for complexity control and reduction
  - Coherence of parts, thus constraining tactical choices
  - Ensure stakeholder concerns are met
- Instrument for planning
  - Definition of a programme of change
- Means of communication
- Framework for development
  - To control the content-coherence of change programs / projects

Architecture

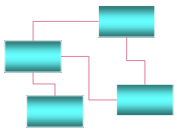
- Architecture:
  - Architecture is the visioning of requirements into a coherent style or structure to help the client make decisions,
  - The architecture of a system is the structure or structures of the system, which comprise services/components, the externally visible properties of those services/components, and the relationships among them.
  - Articulates business direction and focuses technology to support that business direction. It links **vision**, **strategy** and **IT feasibility**, focusing on usability, durability and effectiveness (implementation).
  - Maps a client's requirements to a deployable solution
  - Provides a way to manage complexity and risk, the foundation and justification for IT strategy and provides views of the problem and solution that can be readily understood by all parties - both technical and non-technical.

The key is structure, abstraction and communication

Architecture products

An architecture study delivers two kinds of products

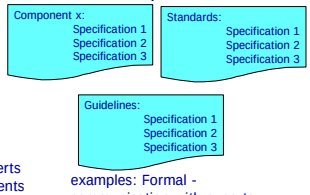
1. Visualisations



example
Formal - communication with experts
Informal - communication with clients

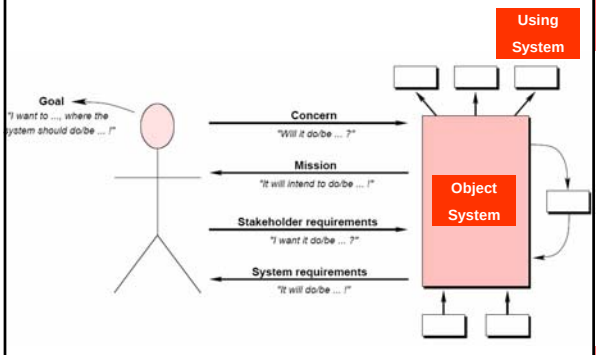
Also for documentation: traceability and maintainability

2. Specifications



examples: Formal - communication with experts (e.g. UML model)

Stakeholders & Concerns



Different stakeholders, different interests

management

IT staff

architects

partners

developers / engineers

sales

end-users

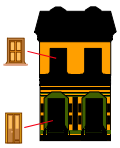
Architecture Is Founded in Principles

- Characteristics of principles
  - Give Direction
  - Making implicit conditions explicit
  - Create delineation
  - Determine Grouping criteria
  - Set priorities
  - Classify ➔ Overview
    - Didn't we forget something?

Example principle

- Principle: Tasks are accessible to automated users in all locations where users need to work to carry out their tasks effectively and efficiently.
- Scope: Work processes
- Motivation: Location independency increases efficiency and flexibility of work processes. If several work locations are available, this allows work to move in case of a (technical) problem.
- Possible refinements:
  - Functionality should not be bound to one particular workstation. It should be made available on groups of workstations, or through a central server or the mainframe.
  - No single installations or stand-alone applications.

Stakeholders, Concerns and Views

Stakeholders	 **Acquirer, Users**	 **Engineers**	 **Builders, Maintenance People**
Concerns	***Fitness for purpose? Usability?***	***Sustainability? Performance?***	***Feasibility? Maintainability?***
View Models			

Views on a system ...

- One system domain
- ... multiple viewers
- ... multiple conceptions
- Multiple views!

Viewpoint

- Perspective:
  - "Perspective - A set of related interests in terms of which viewers may observe a domain"
  - Framework of sub-perspectives
  - Atomic: Business, Information, …
- Viewing method + modeling methods

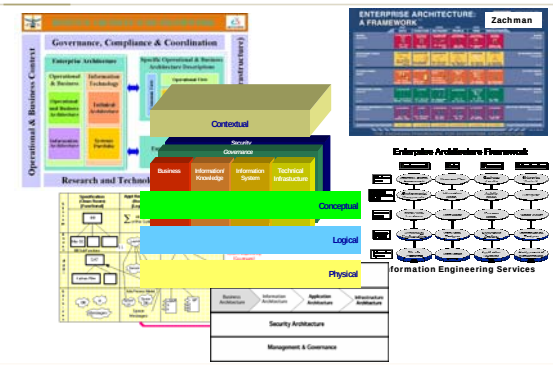
Viewpoint

- IEEE:
  - A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

Frameworks of viewpoints

- Several frameworks of viewpoints exist
  - UML, Zachman, Togaf, ArchiMate, ...
  - Capgemini, Belastingdienst, ...
- A jungle of viewpoints & frameworks
- Which one to use/when ?
  - Subject of research!

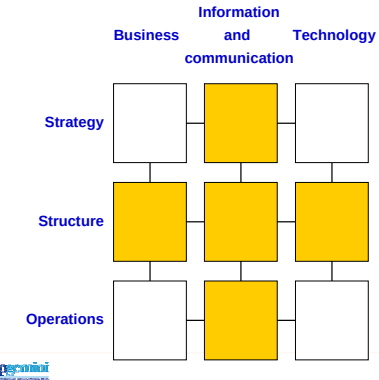
Architecture frameworks



What is a Framework?

- A containment structure:
  - context for model artifacts
  - interconnections between models
  - access to model components
  - model fidelity and consistency

PrimaVera Framework UvA



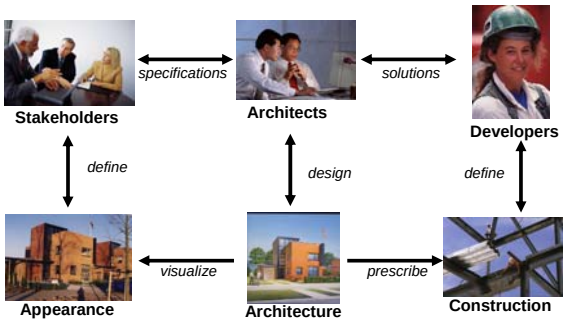
Zachman Framework for Enterprise Architecture

ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why	
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations important to the Business	List of Events Significant to the Business	List of Business Goals/Strat	SCOPE (CONTEXTUAL)
Planner	ENTITY = Class of Business Thing	Function = Class of Business Process	Node = Major Business Location	People = Major Organizations	Time = Major Business Event	Ends/Means=Major Bus. Goal/ Critical Success Factor	Planner
ENTERPRISE MODEL (CONCEPTUAL)	e.g. Semantic Model	e.g. Business Process Model	e.g. Logistics Network	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan	ENTERPRISE MODEL (CONCEPTUAL)
Owner	Ent = Business Entity Rel = Business Relationship	Proc. = Business Process I/O = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	Owner
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model	e.g. "Application Architecture"	e.g. "Distributed System Architecture"	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model	SYSTEM MODEL (LOGICAL)
Designer	Ent = Data Entity Rel = Data Relationship	Proc. = Application Function I/O = User Views	Node = I/S Function (Processor, etc) Link = Line Characteristics	Work = Work Station Role = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion	Designer
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model	e.g. "System Design"	e.g. "System Architecture"	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design	TECHNOLOGY CONSTRAINED MODEL (PHYSICAL)
Builder	Ent = Segment/Table/etc. Rel = Pointer/Key/etc.	Proc. = Computer Function I/O = Screen/Device Formats	Node = Hardware/System Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	Builder
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition	e.g. "Program"	e.g. "Network Architecture"	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
Sub-Contractor	Ent = Field Rel = Address	Proc. = Language Stmt I/O = Control Block	Node = Addresses Link = Protocols	Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step	Sub-Contractor
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

Zachman Institute for Framework Advancement - (810) 231-0531 Copyright - John A. Zachman, Zachman International

Position of an architect

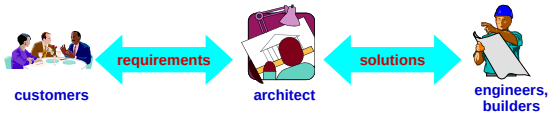


Architecture interacts with Software Design

- Architecture provides the solution structure and guidelines to the Software Design process
- Software Design results can have impact on the architecture:
  - Propose new solutions
  - Discover new technical constraints
  - Discover new guidelines
  - Discover flaws in the structure

Architecture interacts with Software Design

- Architecture provides the solution structure and guidelines to the Software Design process
- Software Design results can have impact on the architecture:
  - Propose new solutions
  - Discover new technical constraints
  - Discover new guidelines
  - Discover flaws in the structure



Architecture versus software development

- Architecture focuses on the Big picture:
  - IT support of the entire business (domain)
  - IT standards and guidelines
- Delivery focuses on ONE system
  - Low cost, Low risk, Quick results
  - Delivery Excellence
- Architecture & Delivery are complementary

- Architects & Engineers have different views on the same problem
  - System Use cases & IS Services for the same problem are NOT equal!
  - IS Services do provide useful input for System Use cases