

*in*Field

Data Management and Archiving

University of California at Santa Barbara, June 24-27, 2008

Analyzing your project in order to build
a database and a use system

Anthony Aristar

- Most linguists build their projects by starting from the lowest end, e.g.:
 - I need to get working! How do I get the answers I need quickly?
 - Or
 - I need to store this data, where can I stick it while I work on the *really* important stuff?
 - This produces systems which:
 - Are in constant flux
 - Can never produce the data in the way that it is really needed.
 - Worst of all, look like they were just thrown together. (Hmmm, they were, weren't they?)

- There's a better way, but it takes some self-discipline to use it.
- That way is to grit your teeth and take the time to analyze all aspects of your project and plan your information system (database and other systems) before you do anything.
- There are techniques that allow you to do that... and they pay off! In computerese, it's called "Information Engineering."

- ... is a formal methodology that is used to create and maintain information systems.
- ... and always starts with the **project model** and works in a *Top-Down* fashion to build supporting data models and process models which best suit your project.
- Basic rule: ***Do the hard parts first!***

*in*Field

The major steps to a good project

Planning

Analysis







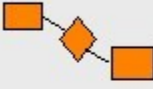
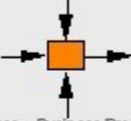

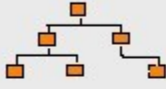

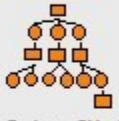
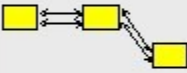
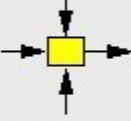
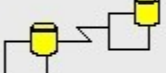
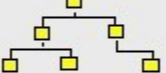

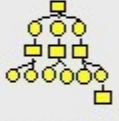
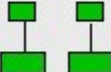

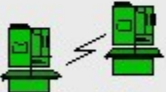
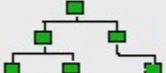

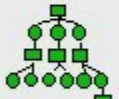






Design

Implementation

- 1987, Zachman, IBM Systems Journal
- 1993, Sowa and Zachman, IBM Systems Journal
- A popular project view of the components of an information system and the process of developing it.
- But equally useful in academic systems because it helps you design an information system of **any** type.
- This is the design that underlies all the systems used by the LINGUIST List.

- This all started because someone noticed that you could build a house or a battleship on time and in budget, but software projects never...
- So IBM brought in a guy named Zachman who analyzed what was going on.
- He noted that the later mistakes were caught, the longer they took to fix, and the more money they cost.

The Full Zachman Framework




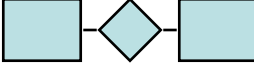
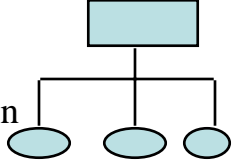
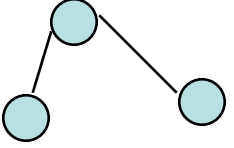
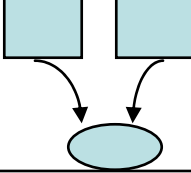
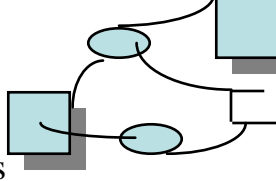
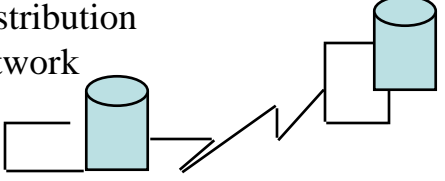
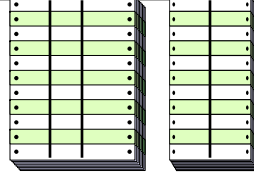
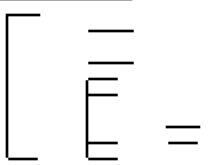
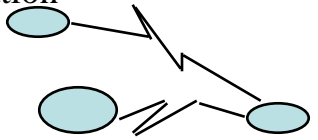
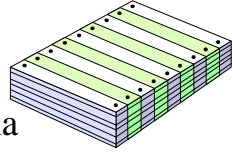
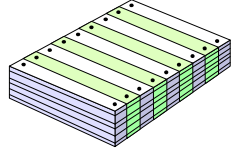
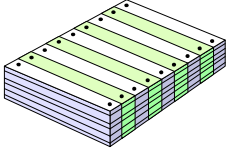
abstractions	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>
SCOPE <i>Planner</i> contextual	List of Things - Important to the Business  Entity = Class of Business Thing	List of Processes - the Business Performs  Function = Class of Business Process	List of Locations - in which the Business Operates  Node = Major Business Location	List of Organizations - Important to the Business  People = Class of People and Major Organizations	List of Events - Significant to the Business  Time = Major Business Event	List of Business Goals and Strategies  Ends/Means=Major Business Goal/Critical Success Factor
ENTERPRISE MODEL <i>Owner</i> conceptual	e.g., Semantic Model  Entity = Business Entity Rel. = Business Relationship	e.g., Business Process Model  Process = Business Process IO = Business Resources	e.g., Logistics Network  Node = Business Location Link = Business Linkage	e.g., Work Flow Model  People = Organization Unit Work = Work Product	e.g., Master Schedule  Time = Business Event Cycle = Business Cycle	e.g., Business Plan  End = Business Objective Means = Business Strategy
SYSTEM MODEL <i>Designer</i> logical	e.g., Logical Data Model  Entity = Data Entity Rel. = Data Relationship	e.g., Application Architecture  Process = Application Function IO = User Views	e.g., Distributed System Architecture  Node = IS Function Link = Line Characteristics	e.g., Human Interface Architecture  People = Role Work = Deliverable	e.g., Processing Structure  Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model  End = Structural Assertion Means = Action Assertion
TECHNOLOGY CONSTRAINED MODEL <i>Builder</i> physical	e.g., Physical Data Model  Entity = Tables/Segments/etc. Rel. = Key/Points/etc.	e.g., System Design  Process = Computer Function IO = Data Elements/Sets	e.g., Technical Architecture  Node = Hardware/System Software Link = Line Specifications	e.g., Presentation Architecture  People = User Work = Screen/Device Format	e.g., Control Structure  Time = Execute Cycle Cycle = Component Cycle	e.g., Rule Design  End = Condition Means = Action
DETAILED REPRESENTATIONS <i>Subcontractor</i> out-of-context	e.g., Data Definition  Entity = Field Rel. = Address	e.g., Program  Process = Language Statement IO = Control Block	e.g., Network Architecture  Node = Addresses Link = Protocols	e.g., Security Architecture  People = Identity Work = Job	e.g., Timing Definition  Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification  End = Sub-condition Means = Step
FUNCTIONING ENTERPRISE	DATA Implementation	FUNCTION Implementation	NETWORK Implementation	ORGANIZATION Implementation	SCHEDULE Implementation	STRATEGY Implementation

A Simplified Model of the Zachman Framework

Data (entities)

Process

Network

	Data (entities)	Process	Network
1	project Scope List of entities important to project 	List of functions project performs 	List of locations in which project operates 
2	project model project entities & their relationships 	Function & process decomposition 	Comms. links between project locations 
3	ISs model Model of project data & their relationships 	Flows between apps. processes 	Distribution network 
4	Technology model Database design 	Process specification 	Configuration design 
5	Technology definition database schema & subschema definition 	Process Specification (e.g. export) 	Configuration definition 
6	Information system data & information	Apps. programs	System configuration <div style="text-align: right;">9</div>

- Zachman modeled his enterprise architecture on the processes involved in building a house.
- You don't just hire a carpenter and give him some nails
- You have at least 6 roles involved in the design of a house

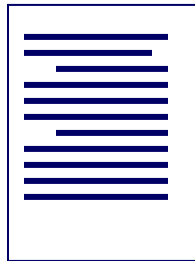
- Six roles involved in the design of a house
 - Owner (defines Project scope)
 - Architect (makes Project model)
 - Designer (makes System model--chooses fittings, materials)
 - Builder (makes Technology model--plans the actual building process)
 - Sub-contractor (implements Technology--installs the plumbing, builds walls, etc.)
 - Resident (makes use of the project)

- How does this model translate into analyzing a linguistic project?
- Six levels at which the project can be analyzed:
 - **Project scope:** a strategic overview including project scope/mission & direction
 - **Project model:** what entities take part in the project and their relationships
 - **Information Systems model:** how this translates into a database design and what forms are necessary in order to access it in the ways needed.
 - **Technology model:** Converts IS model into a design that conforms to the features/constraints of the technology
 - **Technology definition (Programmer):** Converts technology models into statements to generate the actual IS
 - **Information system (User):** Manages, uses and operates the completed IS

- The project can be broken down into 3 parts: Data (entities), Processes, Network (locations)
- Each of these can be analyzed at all 6 levels

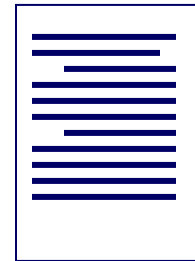
Data

List of entities important to the project



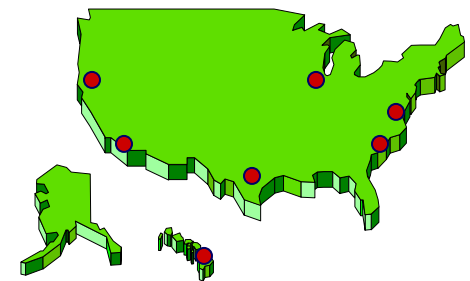
Process

List of processes or functions that the project performs



Network

List of locations in which the project operates



**1. Project Scope
(Overview, goals)**

Data (Entities)

People

- Researcher
- Consultants
- Technology consultant

Data

- Lexical items
- Glosses

Media

- sound files
- video files

Equipment

- computers...

Process

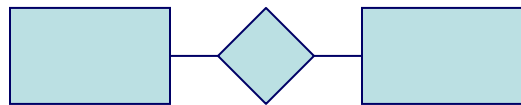
- Collecting Data
- Analyzing data
- Storing data
- Publishing data

Network

- Field setting (village)
- Field setting (town)
- Home office
- University office

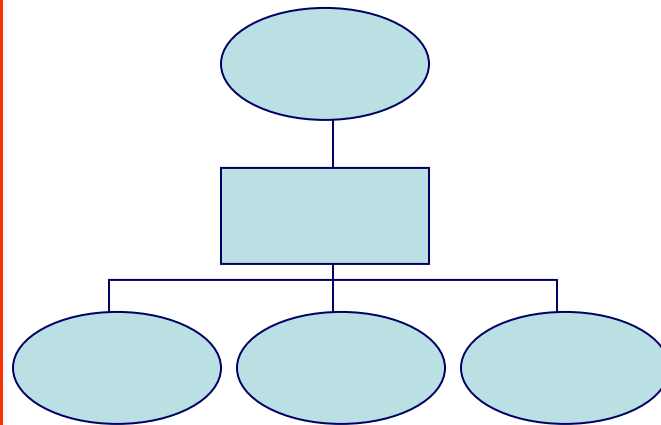
Data

Project entities and their relationships



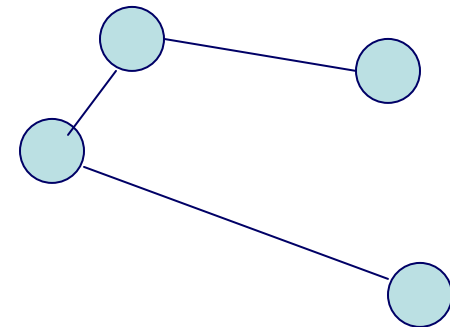
Process

Function and process decomposition



Network

Communications links between project locations



2. Project Model (Architect)

Data (Entities)

Recorders record one or more sound files...

Sound files contain one or more texts

Consultants are linked to sound files

Lexical items appear in one or many texts

Glosses appear on lexical items

video files contain texts

Process

Searches on lexical items need to be made by sub-string search, by whole words

You need to know what files a lexical item appears in, how often they appear, what their context is.

You need to know what texts a consultant provided, and at what location.

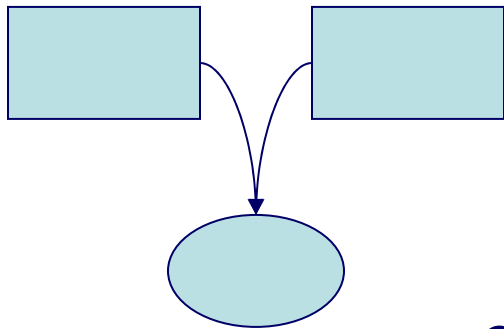
Network

Where are copies of the data kept?

What machines does each kind of file appear on? Where is that machine?

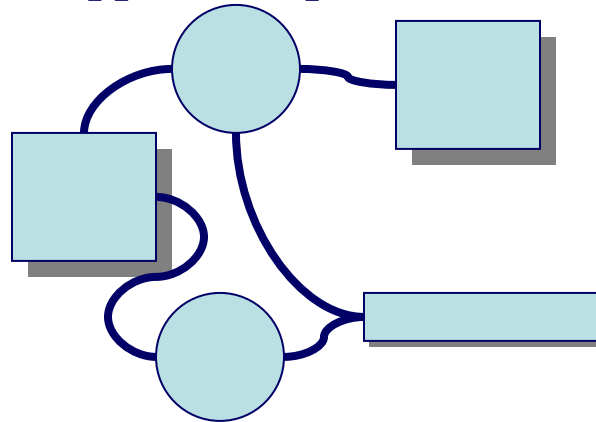
Data

Model of the project data and their relationships (ERD in Database design)



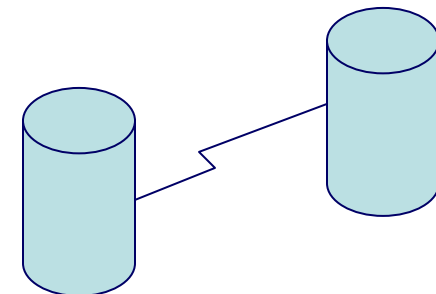
Process

Flows between application processes



Network

Distribution Network



**3. Information System Model
(Designer)**

Data (Entities)

Build database design based on project model: e.g. where lexical item can appear in more than one file, reflect it.

Where consultant provides many texts, reflect it.

Process

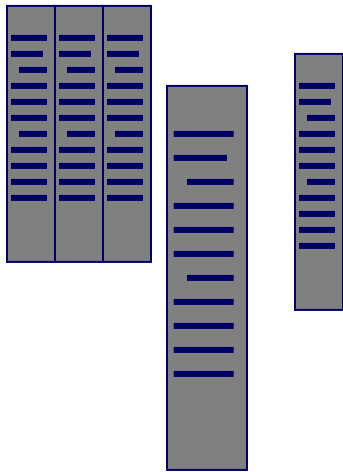
Define the set of forms which will be required to access the data according to the previous two steps, and how they will relate.

Network

Define where this data is to be distributed from.

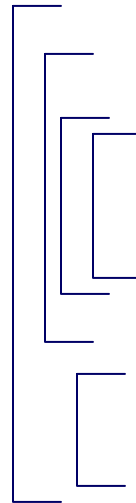
Data

Database Design (logical)



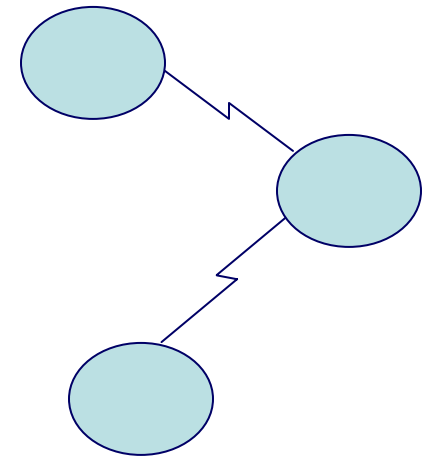
Process

Process specifications



Network

Database Design



**4. Technology Constrained Model
(Builder)**

inField Technology: Documentation project

Data (Entities)

Modify design if necessary to fit technology, assign tasks

Process

Modify design if necessary to fit technology, assign tasks

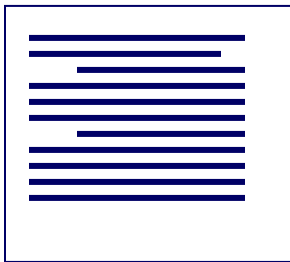
.

Network

If necessary, redesign database to fit.

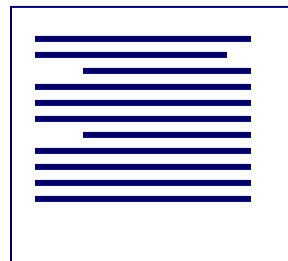
Data

Database Schema
and subschema
definition



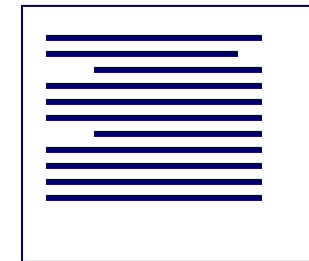
Process

Program Code and
control blocks



Network

Configuration
definition/ Network
Architecture



**5. Technology Definition/
Detailed Representations**

(Programmer)
Data Management & Archiving

inField Technology: Documentation project

Data (Entities)

Instantiate database

Process

Write forms to access
data

.

Network

Instantiate network if
needed.

Data

**Implemented
Database and
information**

Process

**Implemented
Application
Programs**

Network

**Current
System
Configuration**

**6. Functioning project
(User)**

- A fully functional database and access system.

- Understand your project
- Use models to design your system
- Use hierarchical, top-down design, because when these change they force changes everywhere.
- Do the hard parts first
- Focus on high-risk entities first
- State what, not how (called polymorphism in object-oriented programming). State what needs to be done, not how to implement it (e.g. “play sound data”, not “run cd” or “run tape”).
- Allocate each function to only one component: too many cooks spoil the broth. Also, functions should not depend upon one another.
- Group data and behavior
- Prioritize
- Use open standards
- Your interface depends on your design, so design your interface last