Formalization of Zachman Frameworks

Richard Martin
Tinwisle Corporation
Bloomington IN

Edward Robertson
Computer Science Department
Indiana University

I keep six honest serving men
      (They taught me all I knew);
Their names are What and Why and When
      And How and Where and Who.
I send them over land and sea.
      I send them east and west;
But after they have worked for me,
      I give them all a rest.
I let them rest from nine till five,
      For I am busy then,
As well as breakfast, lunch, and tea,
      For they are hungry men.
But different folk have different views,
      I know a person small–
She keeps ten million serving-men,
      Who get no rest at all!
She sends 'em abroad on her own affairs,
      From the second she opens her eyes–
One million Hows, two million Wheres,
      And seven million Whys!

Rudyard Kipling
"The Elephant's Child"
Just So Verses, 1902

Outline – Formalization of Zachman Frameworks

Prolog

Basic Formalism

Excruciating Details

Abbreviations, Constraints & Views

Exercise: IBM ESS Methodology

Prolog

Other Models

Several models by others motivate our approach:

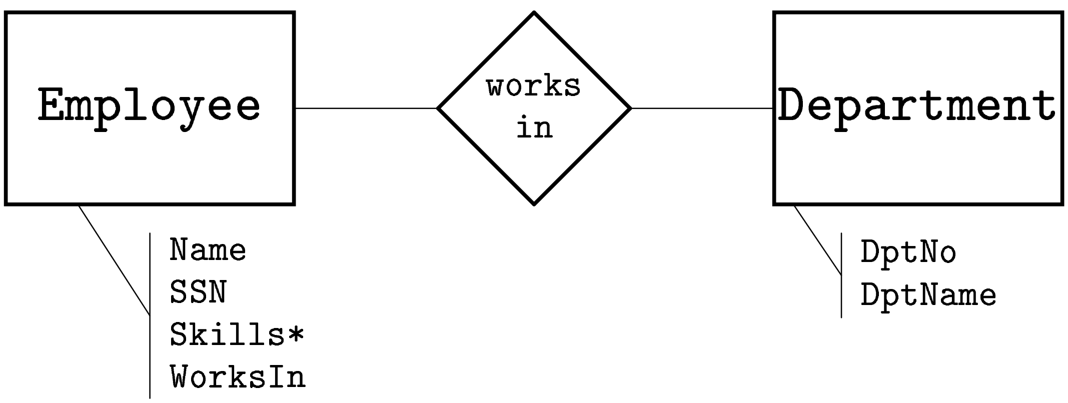
- Zachman's traditional framework architecture – merged with a view of data modeling process
- W. Smith's Meta-ER diagram
- Noriega & Kopko's "Framework use role classification"
- Zachman's "The total picture"

Principles

- Roles are ordered, interrogatives are not
- Entire preceeding role is relevant for each cell of a frame
- Abstraction and detail are not opposite ends of a single spectrum
- Recursion is decomposition and successive refinement, not reoccurrence
- Multiple views are inherent and essential in system modeling and specification

Running Example: an ER Model

- ER model has a well-defined meta-model
- we will use a fragment of ER model for HR system



- We depend on everyone's familiarity as preparation for the example
- ER models are "row 1, column 1" (in our numbering)

Basic Formalism

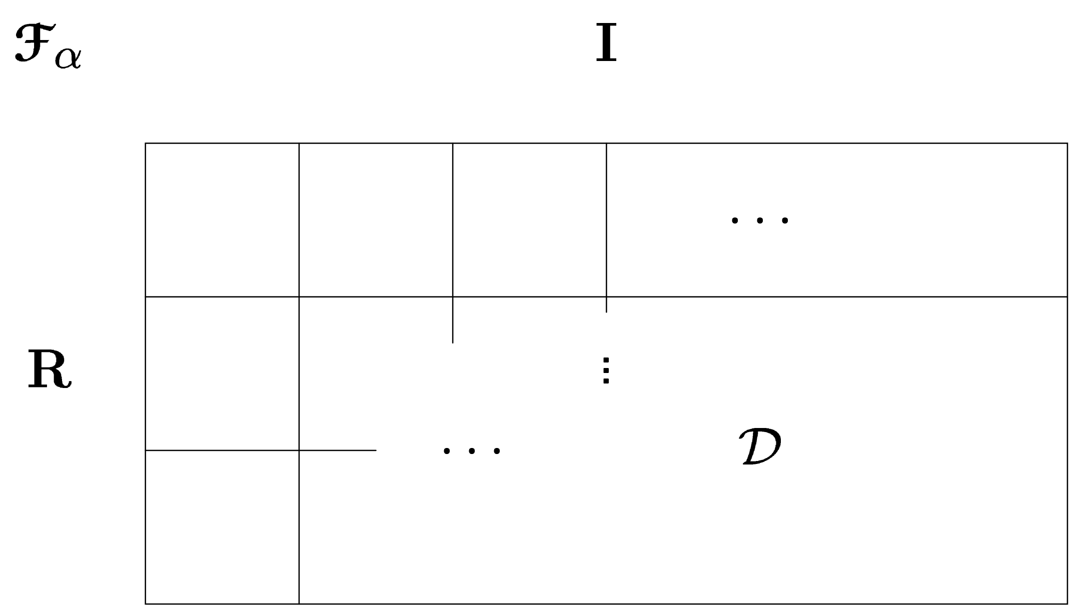
Vocabulary

R $\{r_1, \cdots, r_n\}$, a set of *roles*

I a set of *interrogatives*

$\mathcal{D}$ a finite set, the *descriptors*

$\mathfrak{F}_\alpha$ a *frame*



Roles

$\mathbf{R}$ is ordered

$\mathbf{R}$ typically fixed by methodology

$\mathbf{R}$ specifies rows within a frame

examples of $\mathbf{R}$:

{ "conceptual", "logical", "physical" }

{ "owner", "designer", "builder" }

{ "enterprise", "system", "technology" }

$\mathbf{R}$ is extended with two "contexts":

$\ominus$ before (typically written "context")

$\oplus$ after (typically "out of context")

extended $\mathbf{R}$ is therefore:

$$\{\ominus, r_1, \cdots, r_n, \oplus\}$$

Interrogatives

I is not ordered

I typically fixed by methodology

I specifies columns within a frame

examples of **I**:

{ "what", "how", "where", "who", "when", "why" }

{ "data", "function", "network", "people", "time", "motivation" }

{ "things", "activities", "places", "roles", "events", "rules" }

DESCRIPTORS

$\mathcal{D}$ is unordered

$\mathcal{D}$ localized to application

$\mathcal{D}$ captures "domain knowledge"

examples of $\mathcal{D}$:

{ "Department", "Employee", "Hire", ···}

{ "Accommodate Legacy", "Account", "Actor", "Actual", "Address", "Analyze", "Arbiter", "Architecture", "Architecture Aspect", "Aspect Dependent", "Aspect Difference", "Aspect Integration", "Asset", "Association Notation", "Attitude", "Availability", "Behavior", ···}

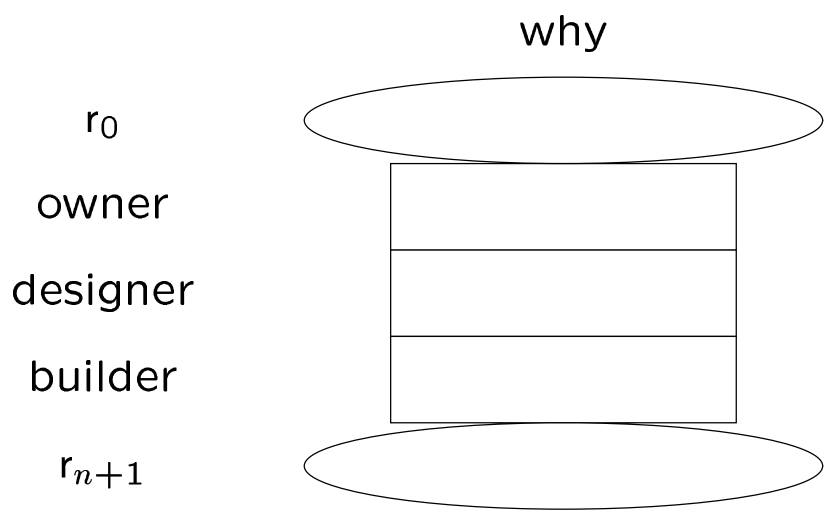
– IBM ESS terminology, 200+ terms

subsets of $\mathcal{D}$ specify the "visible interface" of a frame $\mathfrak{F}_\alpha$:

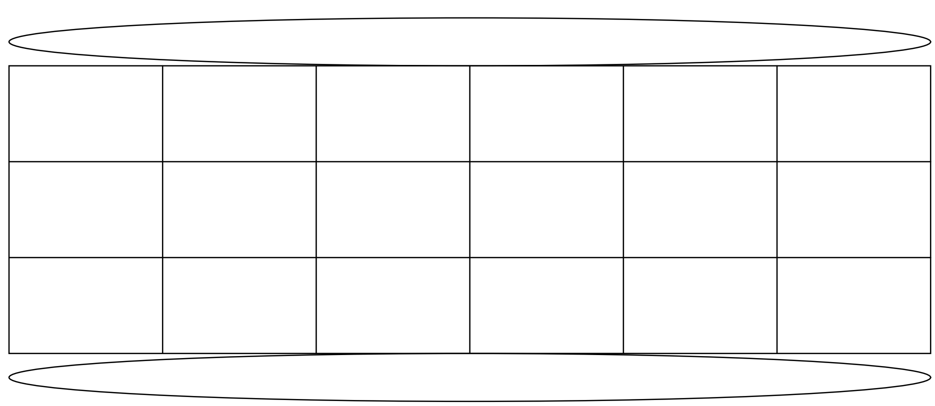
$\mathcal{IC}_\alpha$ is $\mathfrak{F}_\alpha$'s input

$\mathcal{OC}_\alpha$ is $\mathfrak{F}_\alpha$'s output

A Single Column



A Single Frame



Frames and Frameworks

A *framework* $\mathfrak{F}$ is a finite set of *frames*

$$\mathfrak{F} = \{\mathfrak{F}_\alpha : \alpha \text{ is a ``path''}\}$$

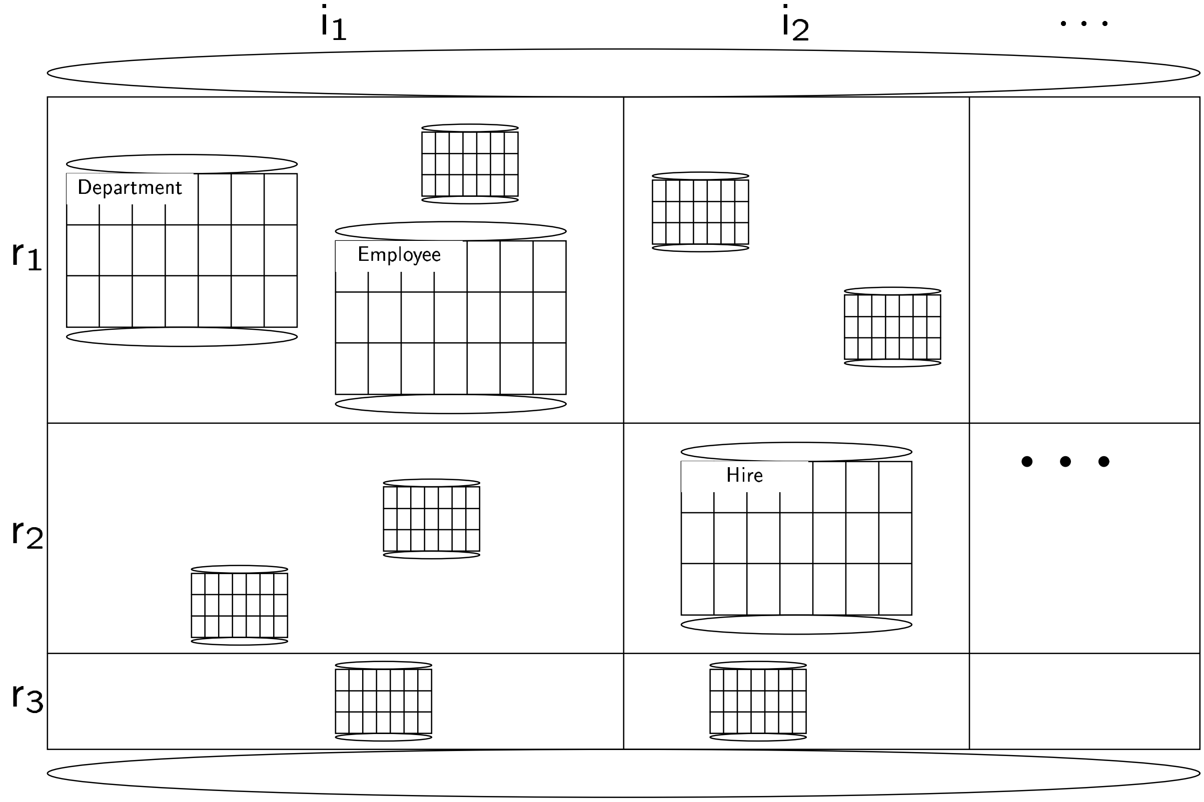
A frame $\mathfrak{F}_\alpha$ is either

- a *branch* frame or
- a *leaf* frame

A "path" is typical of labeling any hierarchical structure, just as "`USA.Illinois.Cook.Chicago`" would label a governmental unit

Branch Frame Structure

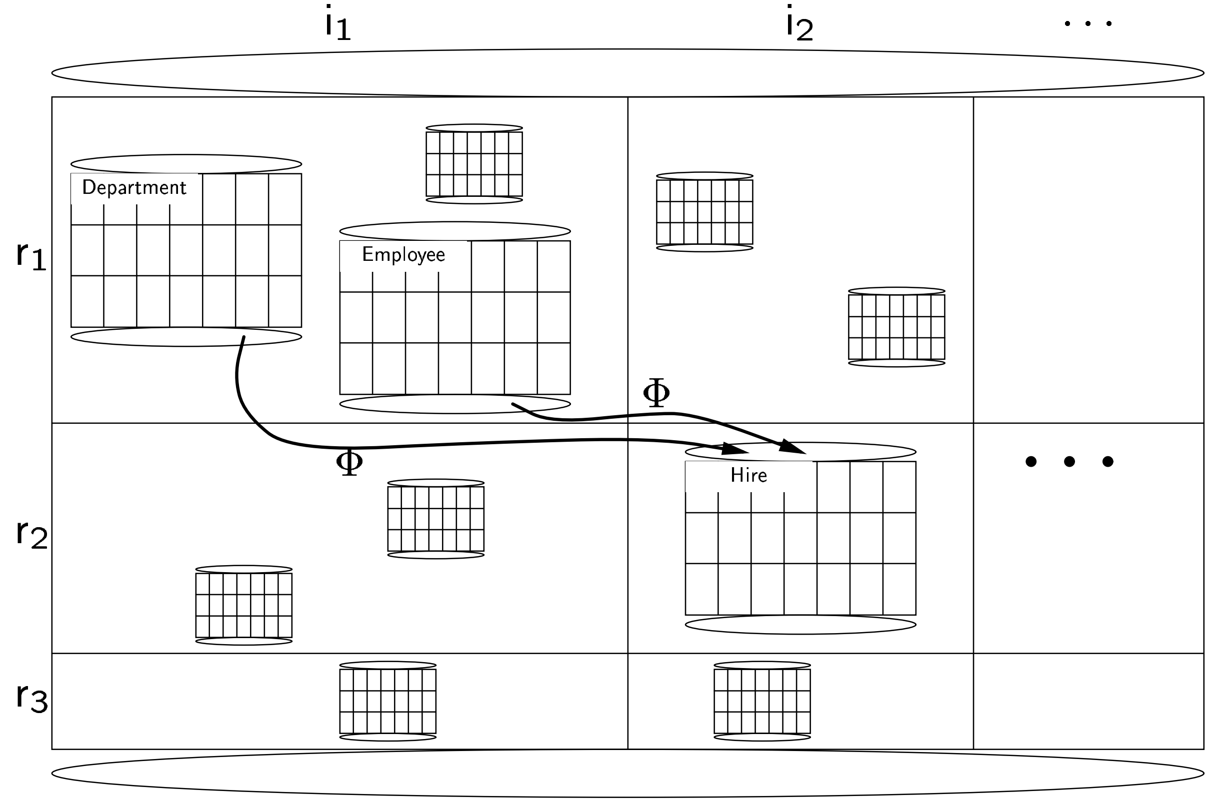
$\mathbf{R} \times \mathbf{I} \times \mathcal{D}$ indexes subframes



- typically $\mathbf{R}$ and $\mathbf{I}$ are implicit

Branch Frame Interconnections

Φ_α connects subframes



- Φ connects
  - to next role
  - from output to input

Leaf Frames

"Hooks" for components from the modeled world

Only the interface of these components is specified

- mechanism is type signature

- may connect to individuals or sets

The Heart of the Meta

- real data is from the real world
- the structure we impose on the real world is the model
- the methodology reflects the structure we impose on models, hence it is a meta-model
- the framework reflects the structure we impose on methodologies, hence it is a meta-meta-model

Labels and Paths

edge labels:

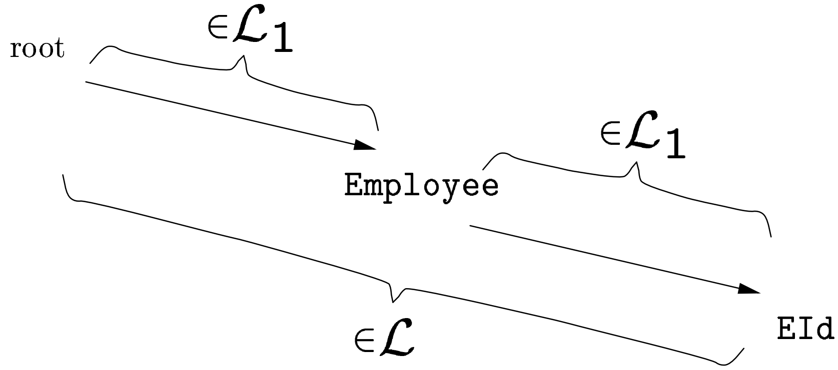
$\mathcal{L}_1$ $\{\langle r, i, d\rangle : r \in \mathbf{R},\ i \in \mathbf{I},\ \&\ d \in \mathcal{D}\}$

path labels:

$\mathcal{L}$ defined by BNF $\mathcal{L} := \epsilon | \mathcal{L}\,\mathcal{L}_1$

(ϵ denotes the empty path)

$\mathcal{PATHS}_{\mathfrak{F}}$ $\{\alpha : \mathfrak{F}_\alpha$ is defined$\}$ for a framework $\mathfrak{F}$



Frames

branch frames:

$\mathfrak{F}_\alpha \quad \langle \mathcal{IC}_\alpha, \mathcal{OC}_\alpha, \quad \mathcal{SF}_\alpha, \quad \Phi_\alpha \rangle$

leaf frames:

$\mathfrak{F}_\alpha \quad \langle \mathcal{IC}_\alpha, \mathcal{OC}_\alpha, \quad \mathfrak{S}_\alpha \rangle$

where

$\mathcal{IC}_\alpha \quad \subseteq \mathcal{D}$

$\mathcal{OC}_\alpha \quad \subseteq \mathcal{D}$

$\left.\begin{matrix}\mathcal{EOC}_{\alpha,r} \\ \mathcal{EIC}_{\alpha,r}\end{matrix}\right\} \quad \subset \mathcal{D}$ restricted to row r

$\mathcal{SF}_\alpha \quad : \mathbf{R} \times \mathbf{I} \times \mathcal{D} \rightarrow \mathfrak{F} \cup \mathfrak{VF}$

$\Phi_\alpha \quad \subseteq \bigcup_{r \in \{\ominus\} \cup \mathbf{R}} (\mathcal{EOC}_{\alpha,r} \times \mathcal{EIC}_{\alpha,r'})$

$Types \quad \mathcal{D} \cup \{\texttt{SET OF } d : d \in \mathcal{D}\}$

$\mathfrak{S}_\alpha \quad : \mathcal{D} \rightarrow \bigcup_{n \in \mathbb{N}} {Types_\alpha}^n$

Abbreviations, Constraints & Views

Abbreviations

Goal: use abbreviations to facilitate formalism, providing

- familiarity
- uniformity
- structure

- a single name $\mathcal{N}$ can abbreviate
  - a path: $\mathcal{N} \rightarrow \langle r, i, d\rangle\langle r', i', d'\rangle$
  - $\mathbf{R} \times \mathbf{I}$ cell coordinates: $\mathcal{N} \rightarrow \langle r, i, \ \cdot \ \rangle$
  - a path template with multiple substitutions: $\mathcal{N} \rightarrow \langle r, i, \ \cdot \ \rangle\langle r', i', \ \cdot \ \rangle$
- use to shorten complicated paths
- envision implementation with macro preprocessor

Abbreviations in ER Example

- in top-level frame, $\langle r_1, i_1, \ \cdot\ \rangle$ subframe (or $\langle$conceptual, data, $\cdot$ $\rangle$ subframe) is commonly called an "Entity" or "Relationship"

→ so saying "Entity abbreviates r_1, i_1" means that "Entity.`Department`" abbreviates $\langle r_1, i_1,$ "`Department`" $\rangle$

- in $\langle r_1, i_1, \ \cdot\ \rangle$ frame, a $\langle r_1, i_1, \ \cdot\ \rangle$ subframe is commonly called an "Attribute"

→ so "Entity.Attribute includes `Employee.SSN`" means $\langle r_1, i_1,$ `Employee`$\rangle.\langle r_1, i_1,$ `SSN`$\rangle$ is a valid path

Constraints

Goal: use restrictions on formal relations ($\mathfrak{F}$, $\mathcal{SF}$, Φ, $\cdots$) to:

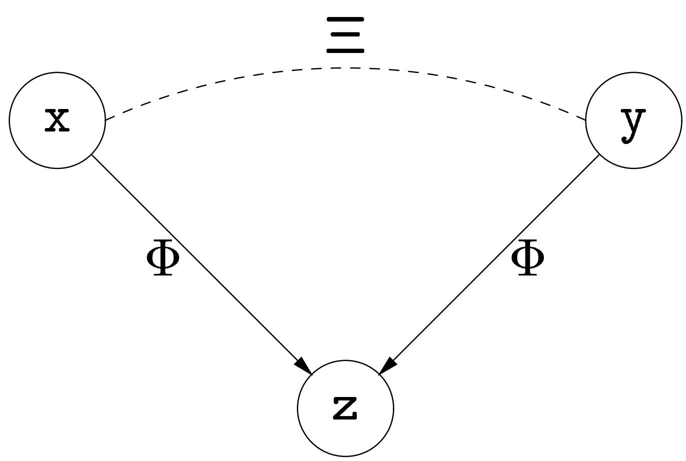
- insure consistency (interior)
- enforce business rules (leaves)

- analogous to foreign key constraints in RDB implementation
- abbreviations constrain legal path labels to provide a methodology's structure

Consistency Constraint Example

Constraint: if Φ maps both $\mathtt{x}$ and $\mathtt{y}$ to $\mathtt{z}$, then $\mathtt{x}$ and $\mathtt{y}$ are consistent

Enforce with new equivalence relationship Ξ



Constraints in ER Example

- multivalued attributes, such as `Skill*`, have proper implementation

- foreign key constraint, such as every employee works in a valid department

$\longrightarrow$ `SET OF Employee.WorksIn.Value`
$\subseteq$ `SET OF Department.DptNo.Value`

Views

Goal: rearrange framework components in order to

- focus on certain elements
- match different perspectives
- validate structure

- achieve this by rewriting path labels
  - omit segments
  - rearrange segments

- full tree rewriting too powerful

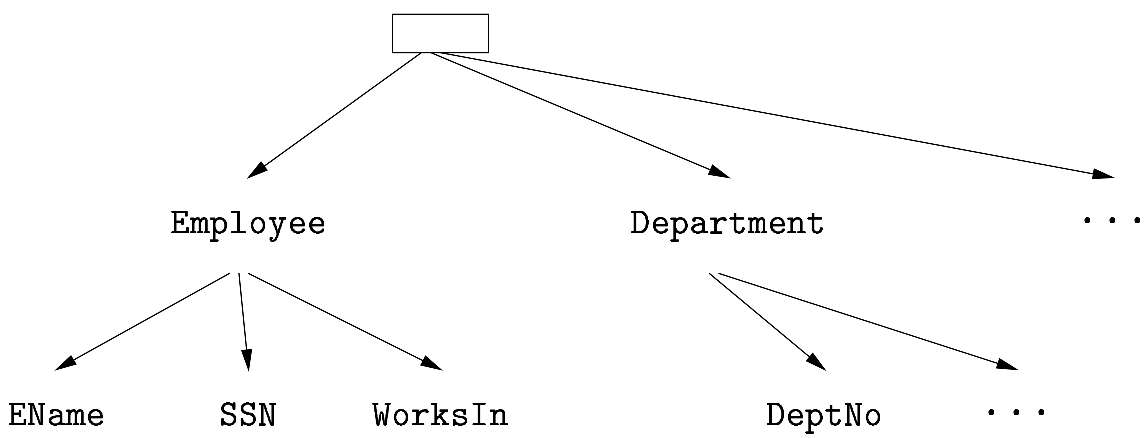
- more structural than semantic
  - difficult to carry along Φ

Views in ER Example

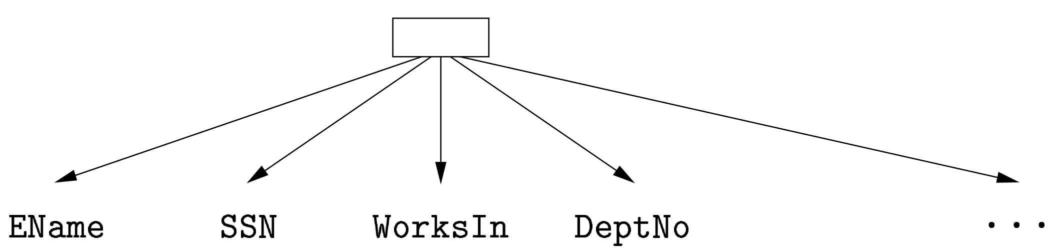
Form data dictionary using rule

$$\langle r_1, i_1, X\rangle.\langle r_1, i_1, Y\rangle \Rightarrow \langle r_1, i_1, Y\rangle$$

so



becomes



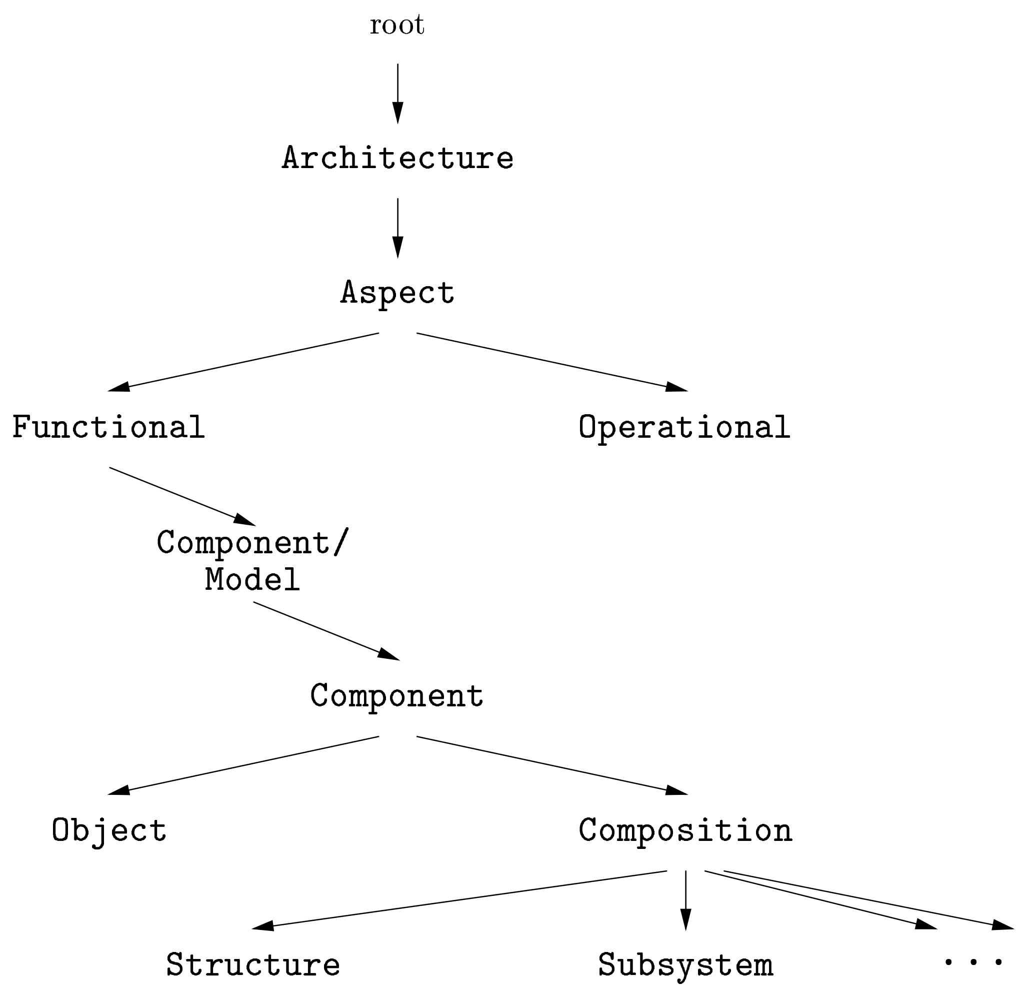
Exercise: IBM ESS Methodology

ESS Background

Goal: validate formalism with application to a real model

- begin with IBM's Enterprise Solutions Structure (*IBM Systems Journal*, 1999)
- applied Doug McDavid's semantic analysis technique to obtain vocabulary
- mapped into our formalism

Fragment of ESS Meta-hierarchy



Note: **R** and **I** indices omitted

ESS Equivalences for I

what	$\approx$	resource
how	$\approx$	behavior
where	$\approx$	location
who	$\approx$	delivery
when	$\approx$	______
why	$\approx$	drivers

Vocabulary for One Frame

Composition – $\mathfrak{F}_\alpha$

what	how	where
Structure	Dynamic_Behavior Enablement	Subsystem
Object_State Partitions Allocations	Object_Interface Optimizing Templates	Grouping Overlap Platform_ Span
UML_Class_ Diagram	Component_ Interaction_ Diagram	UML_Deployment_ Diagram

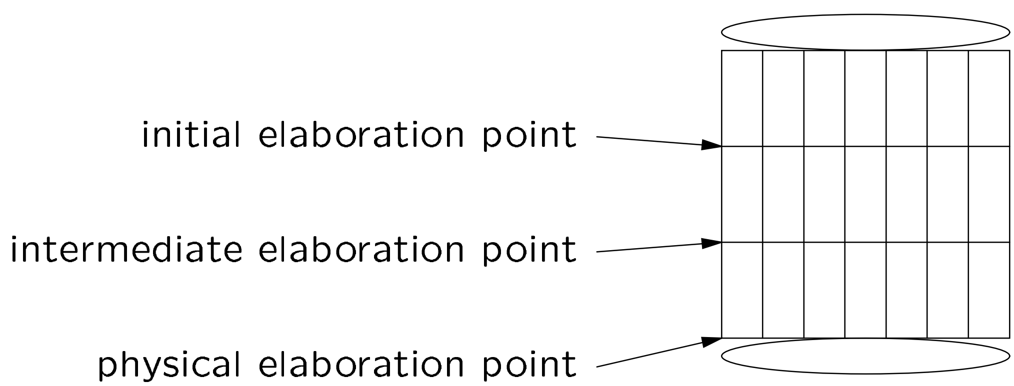
path $\alpha = \langle$conceptual, what, **Architecture**$\rangle$.
$\langle$conceptual, what, **Aspect**$\rangle$.
$\langle$logical, how, **Functional**$\rangle$.
$\langle$logical, what, **Component_Model**$\rangle$.
$\langle$physical, what, **Component**$\rangle$.
$\langle$logical, how, **Composition**$\rangle$

Vocabulary for One Frame – continued

who	when	why
Collaborations Role_Player	Sequence	Use_Case Situation Purpose
Performing_Component Requesting_Component Workflow	Interactions Operation_Sequence Task_Sequence	Interface_Use Service_Level Commitment
UML_Collaboration_Diagram	UML_ Sequence_Diagram	UML_ Scenario

Elaboration Points

ESS makes heavy use of $\mathcal{EOC}$ during modeling process



Future Work

- gain experience
- semantics
  - make use of Φ less awkward
- constraints
  - extend concepts to capture different flavors of frameworks
  - more rigorous formal characterization
- views
  - formalize and investigate formal properties
  - better ways for handling ambiguities
- methodologies & toolkits

Technical report available at

`http://www.cs.indiana.edu/database/Publications/TR522.html`