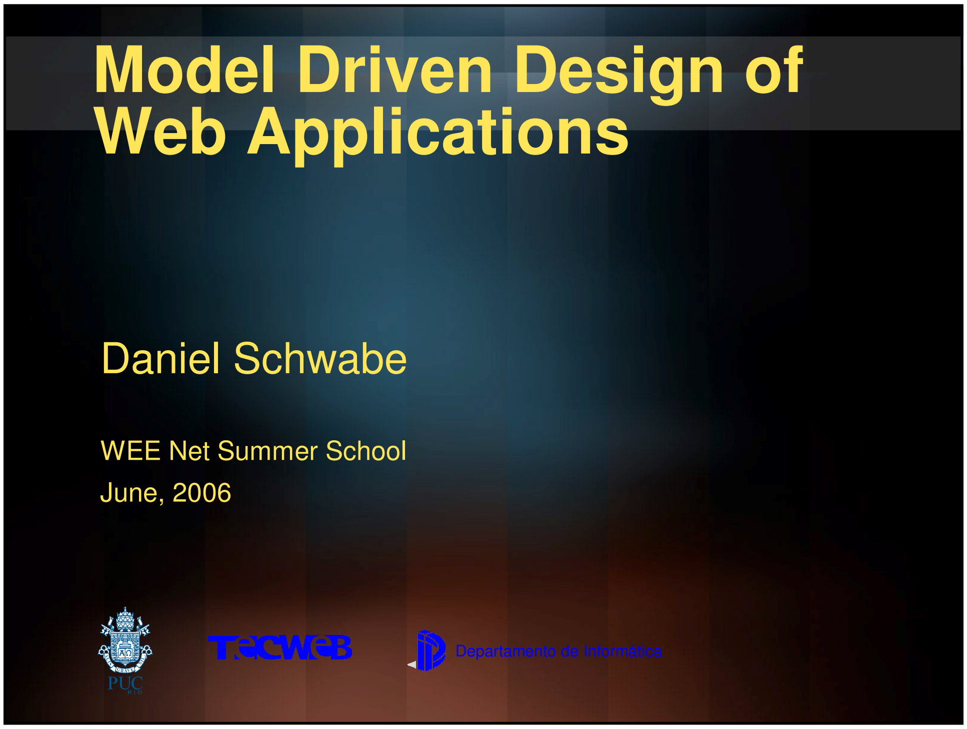


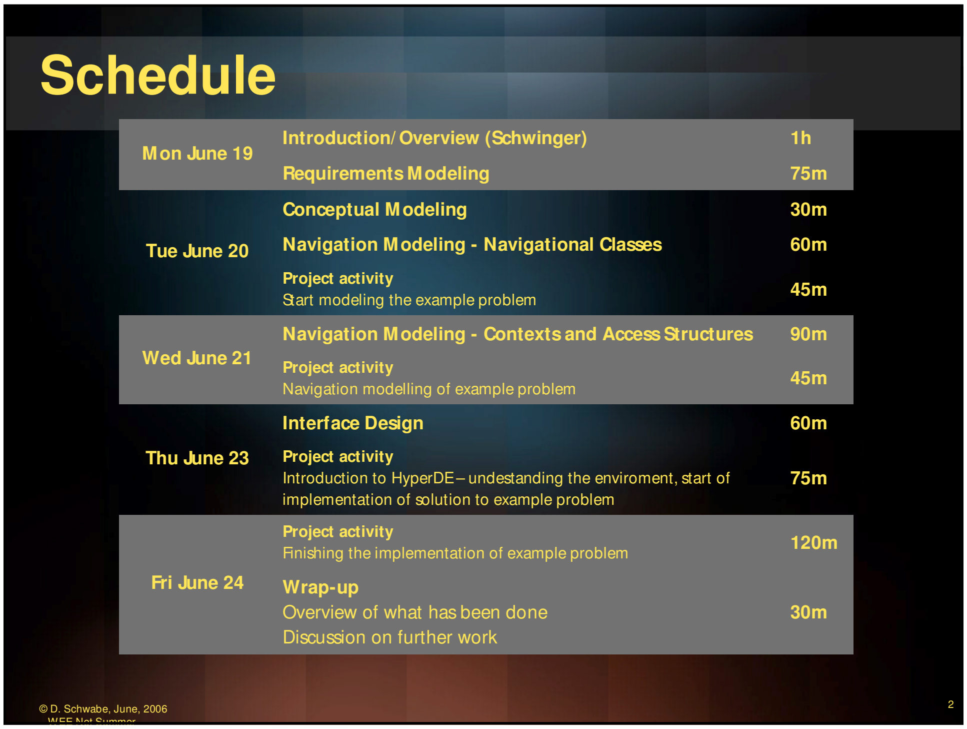
Model Driven Design of Web Applications

Daniel Schwabe

WEE Net Summer School

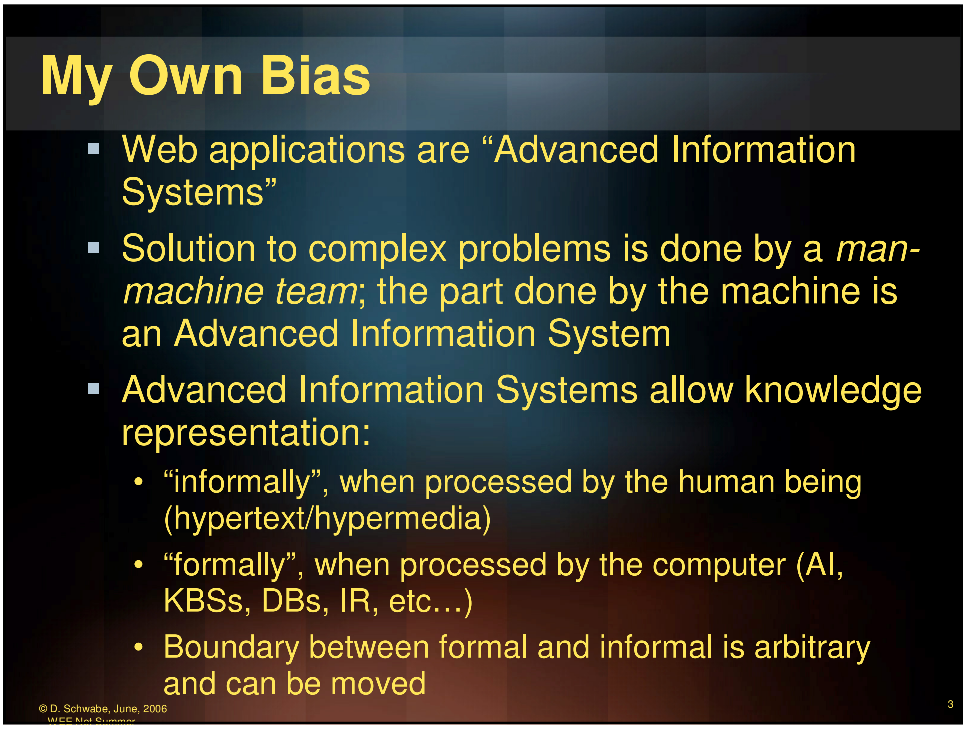
June, 2006

Departamento de Informática



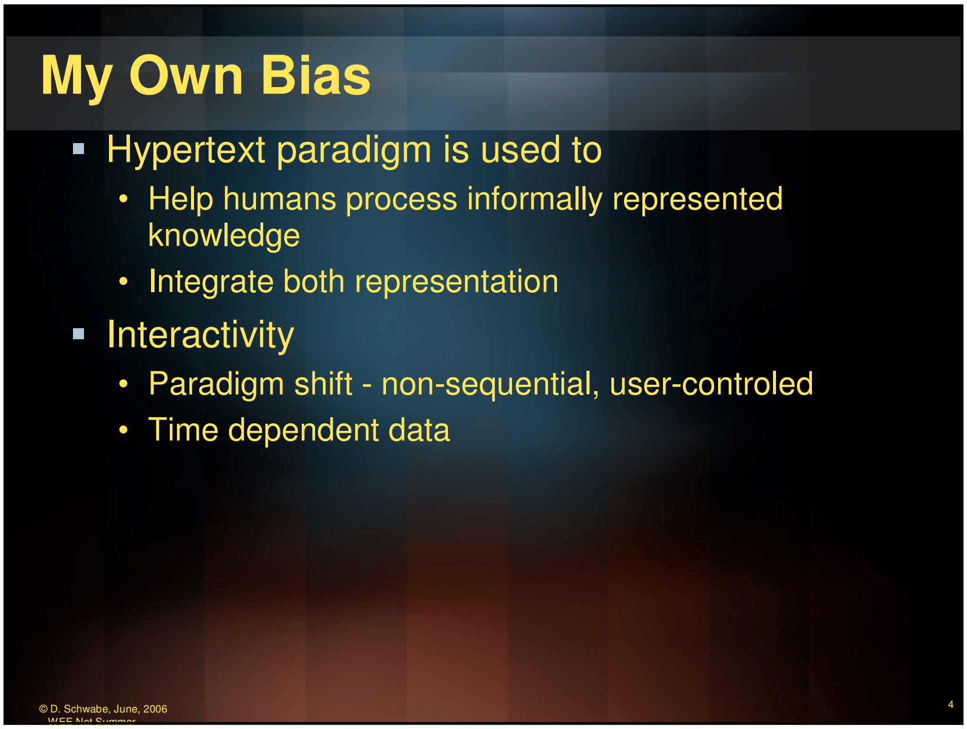
Schedule

Mon June 19	**Introduction/ Overview (Schwinger)**	1h
	Requirements Modeling	75m
Tue June 20	**Conceptual Modeling**	30m
	Navigation Modeling - Navigational Classes	60m
	Project activity Start modeling the example problem	45m
Wed June 21	**Navigation Modeling - Contexts and Access Structures**	90m
	Project activity Navigation modelling of example problem	45m
Thu June 23	**Interface Design**	60m
	Project activity Introduction to HyperDE – undestanding the enviroment, start of implementation of solution to example problem	75m
Fri June 24	**Project activity** Finishing the implementation of example problem	120m
	Wrap-up Overview of what has been done Discussion on further work	30m



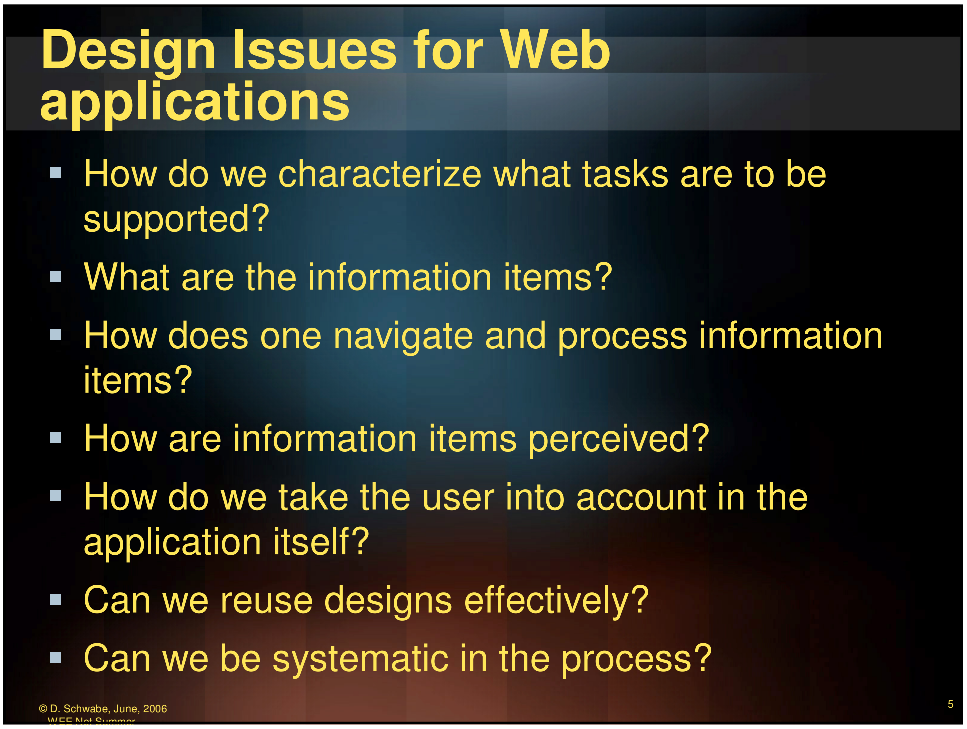
My Own Bias

- Web applications are “Advanced Information Systems”
- Solution to complex problems is done by a *man-machine team*; the part done by the machine is an Advanced Information System
- Advanced Information Systems allow knowledge representation:
  - “informally”, when processed by the human being (hypertext/hypermedia)
  - “formally”, when processed by the computer (AI, KBSs, DBs, IR, etc…)
  - Boundary between formal and informal is arbitrary and can be moved



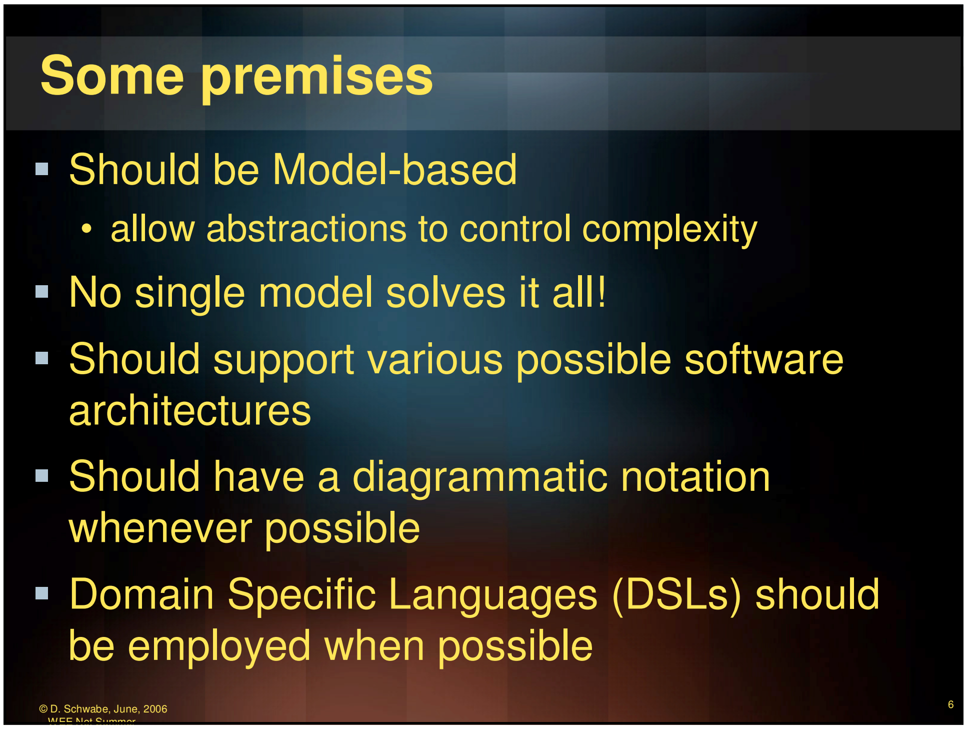
My Own Bias

- Hypertext paradigm is used to
  - Help humans process informally represented knowledge
  - Integrate both representation
- Interactivity
  - Paradigm shift - non-sequential, user-controled
  - Time dependent data



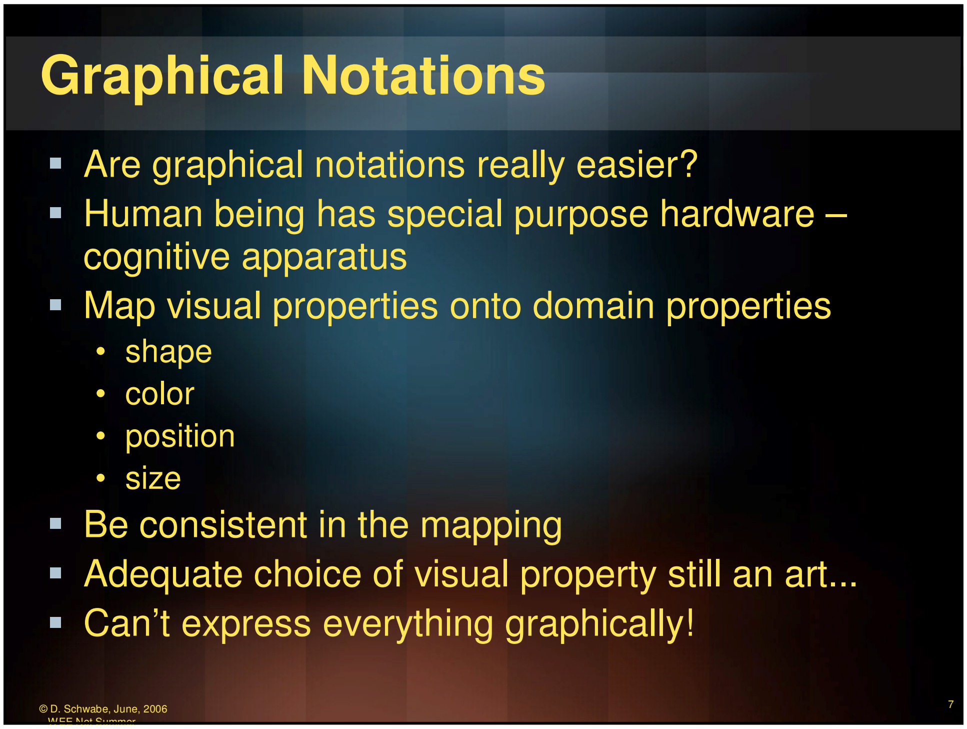
Design Issues for Web applications

- How do we characterize what tasks are to be supported?
- What are the information items?
- How does one navigate and process information items?
- How are information items perceived?
- How do we take the user into account in the application itself?
- Can we reuse designs effectively?
- Can we be systematic in the process?



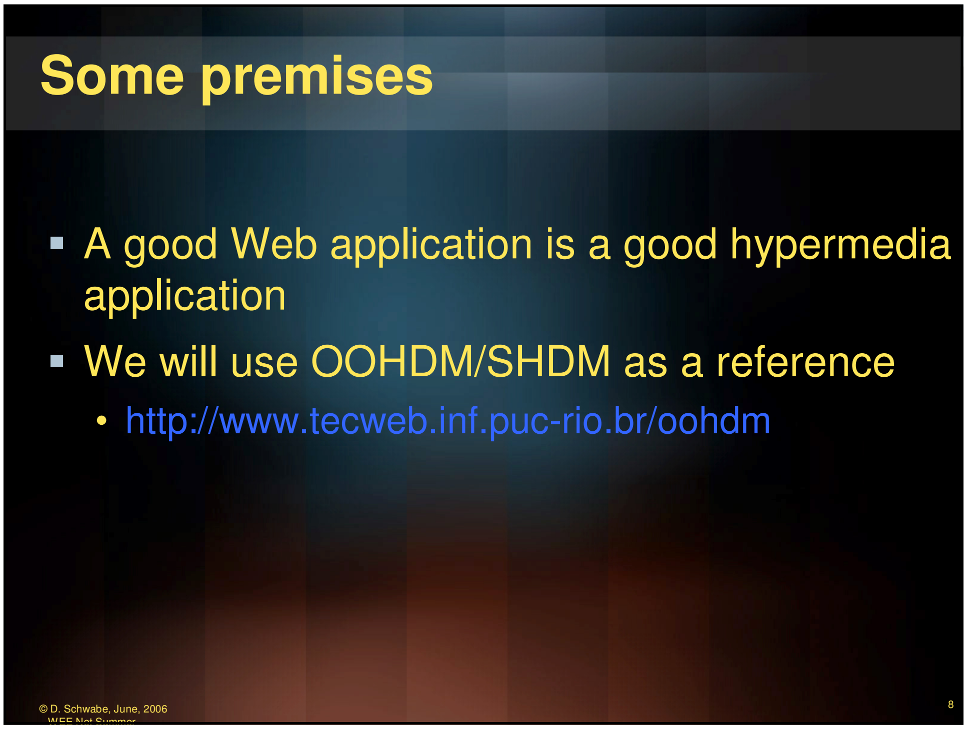
Some premises

- Should be Model-based
  - allow abstractions to control complexity
- No single model solves it all!
- Should support various possible software architectures
- Should have a diagrammatic notation whenever possible
- Domain Specific Languages (DSLs) should be employed when possible



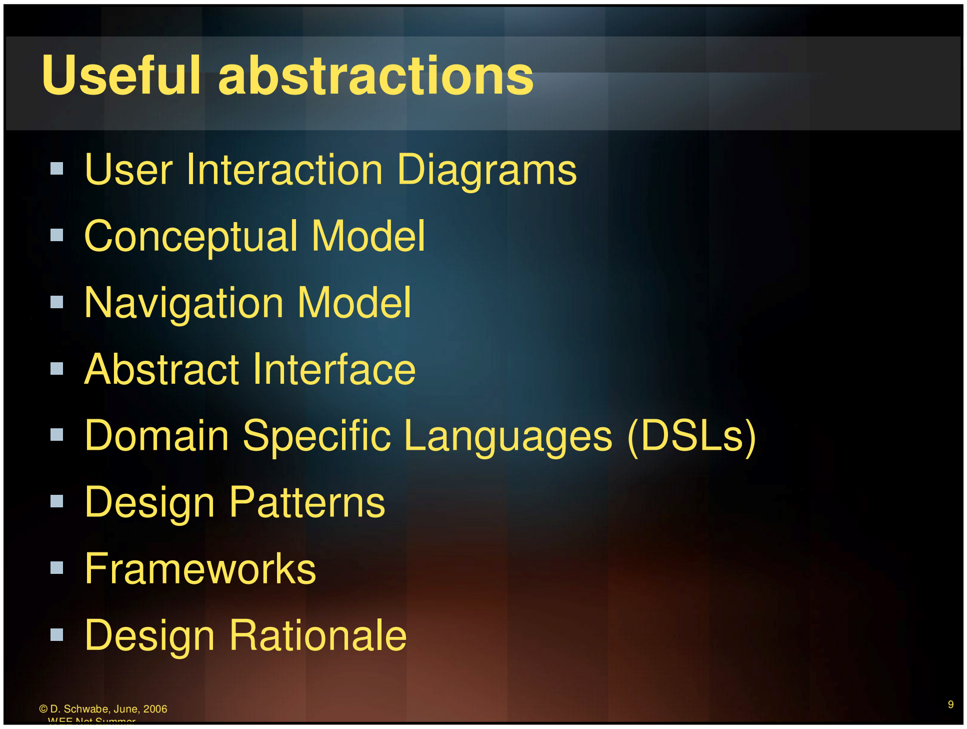
Graphical Notations

- Are graphical notations really easier?
- Human being has special purpose hardware – cognitive apparatus
- Map visual properties onto domain properties
  - shape
  - color
  - position
  - size
- Be consistent in the mapping
- Adequate choice of visual property still an art...
- Can't express everything graphically!



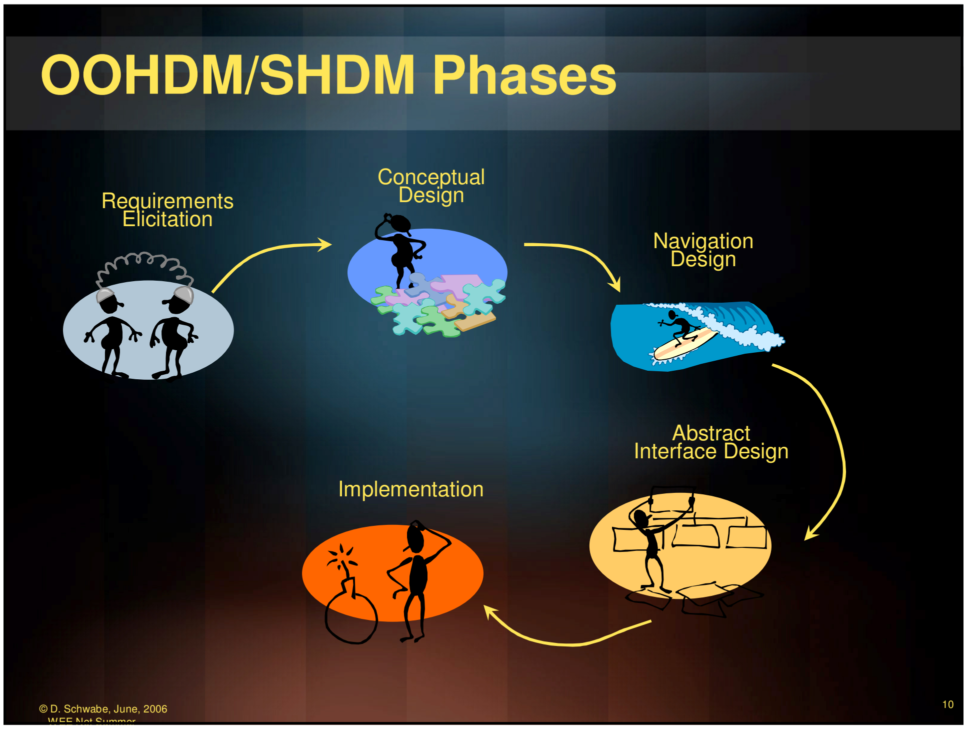
Some premises

- A good Web application is a good hypermedia application
- We will use OOHDM/SHDM as a reference
  - http://www.tecweb.inf.puc-rio.br/oohdm



Useful abstractions

- User Interaction Diagrams
- Conceptual Model
- Navigation Model
- Abstract Interface
- Domain Specific Languages (DSLs)
- Design Patterns
- Frameworks
- Design Rationale



OOHDM/SHDM Phases

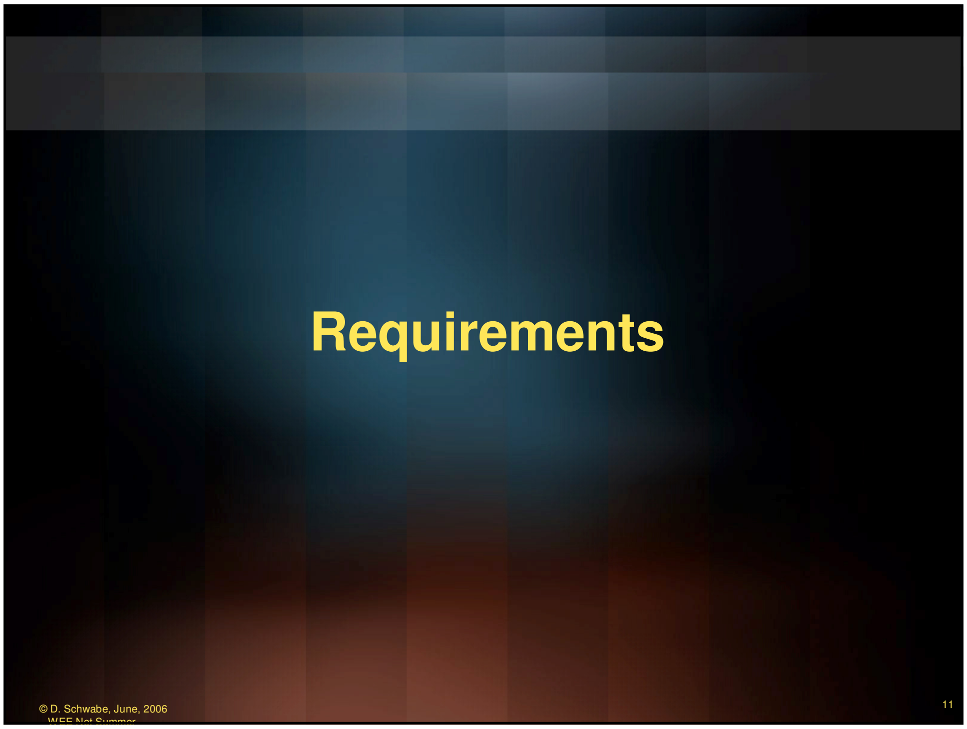
Requirements Elicitation

Conceptual Design

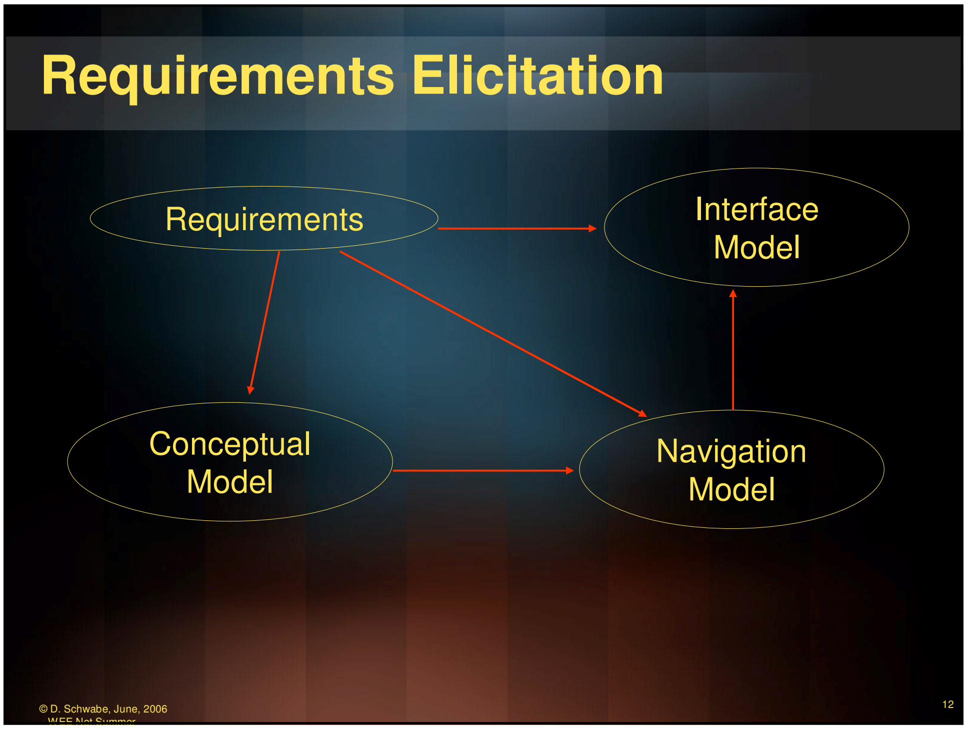
Navigation Design

Abstract Interface Design

Implementation



Requirements



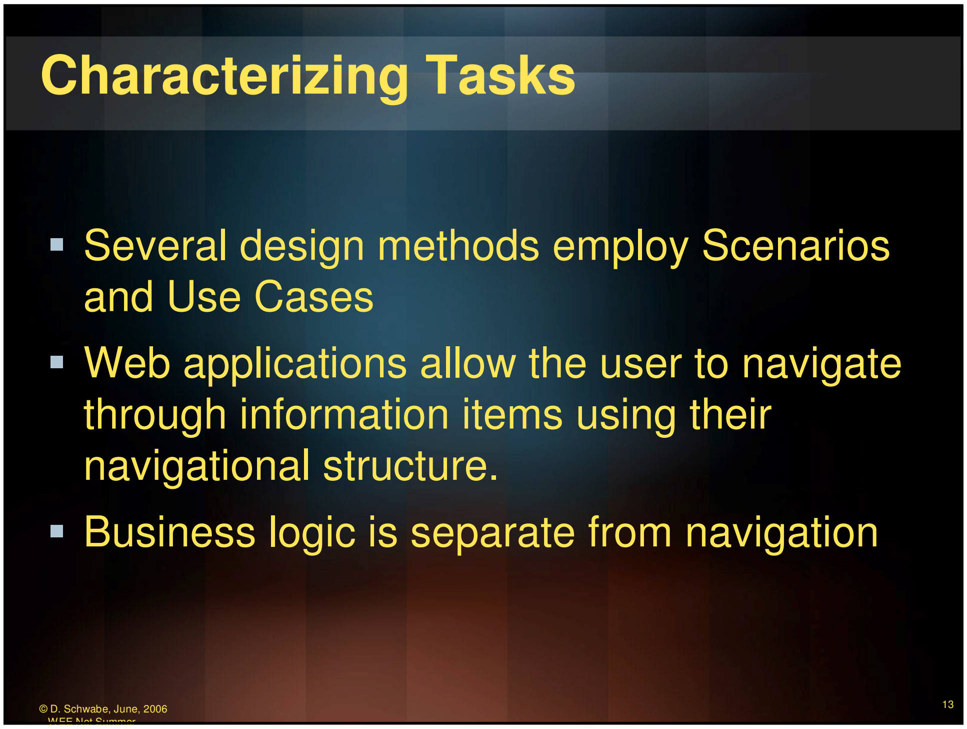
Requirements Elicitation

Requirements

Interface Model

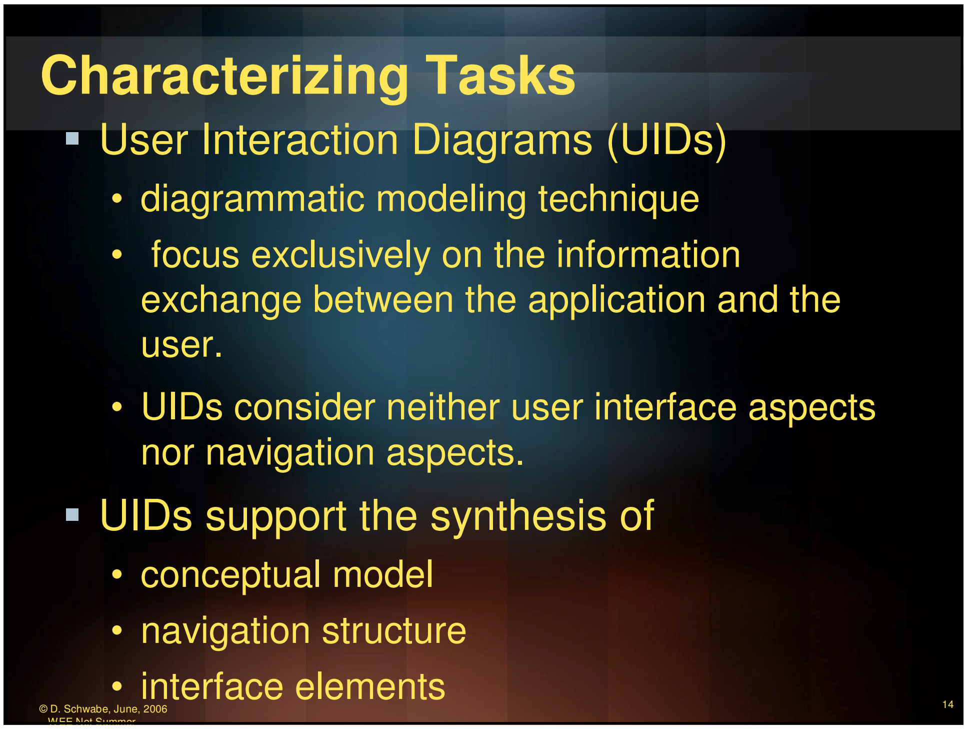
Conceptual Model

Navigation Model



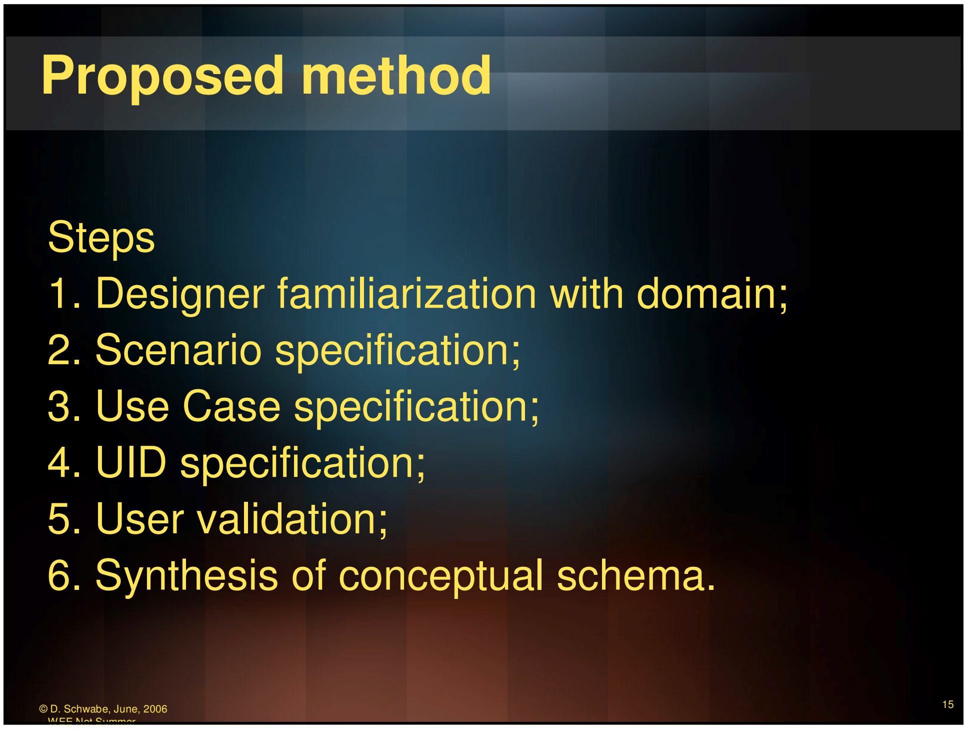
Characterizing Tasks

- Several design methods employ Scenarios and Use Cases
- Web applications allow the user to navigate through information items using their navigational structure.
- Business logic is separate from navigation



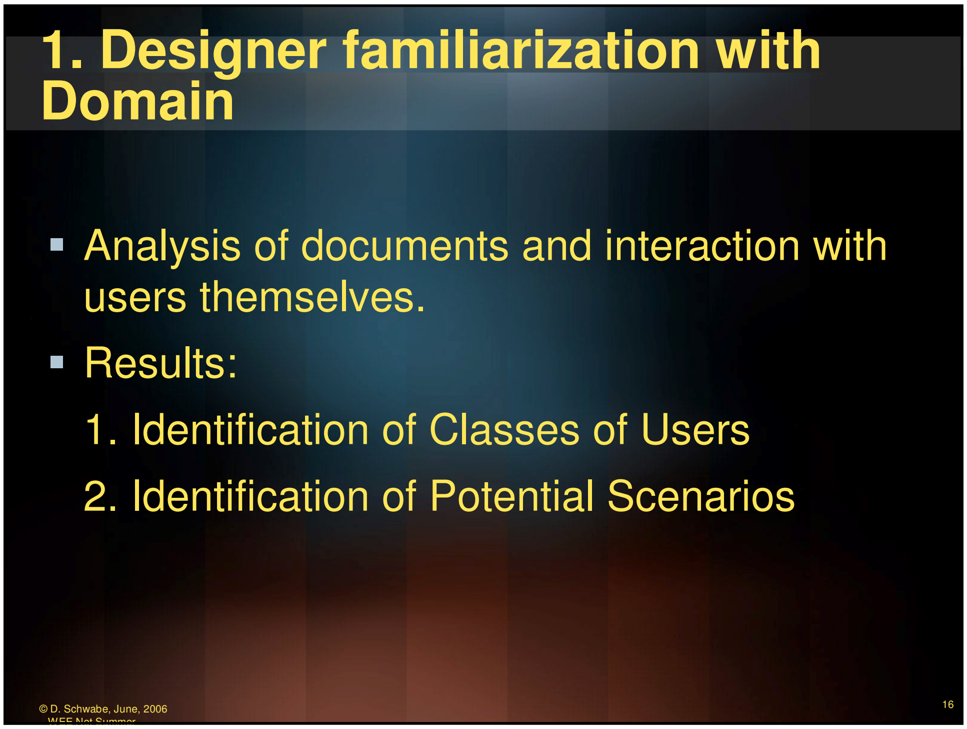
Characterizing Tasks

- User Interaction Diagrams (UIDs)
  - diagrammatic modeling technique
  - focus exclusively on the information exchange between the application and the user.
  - UIDs consider neither user interface aspects nor navigation aspects.
- UIDs support the synthesis of
  - conceptual model
  - navigation structure
  - interface elements



Proposed method

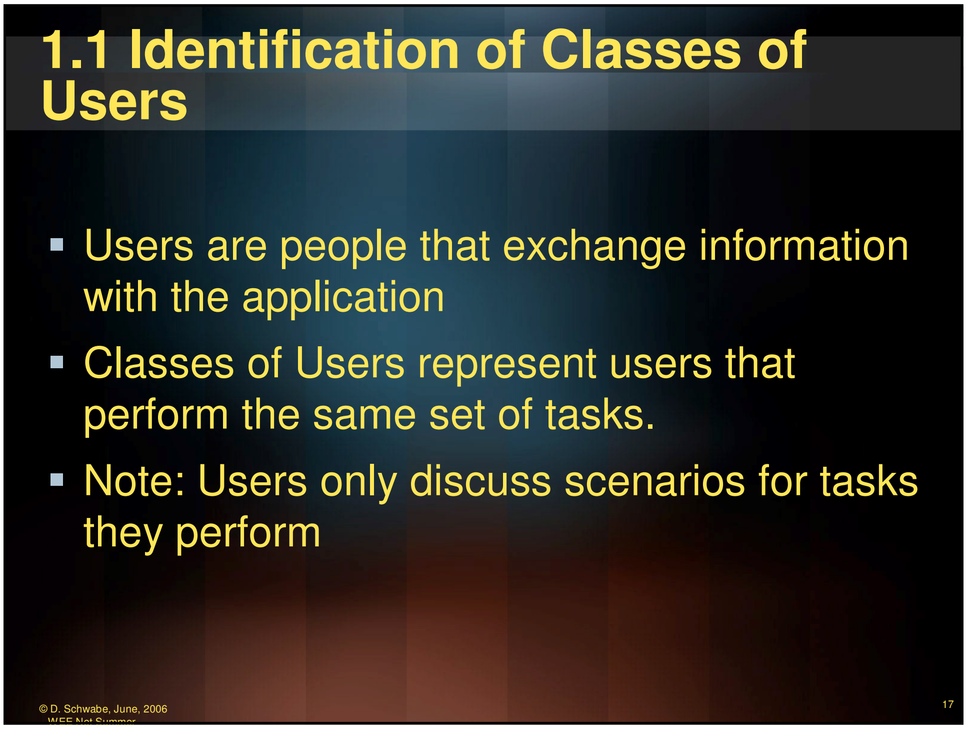
Steps
1. Designer familiarization with domain;
2. Scenario specification;
3. Use Case specification;
4. UID specification;
5. User validation;
6. Synthesis of conceptual schema.



1. Designer familiarization with Domain

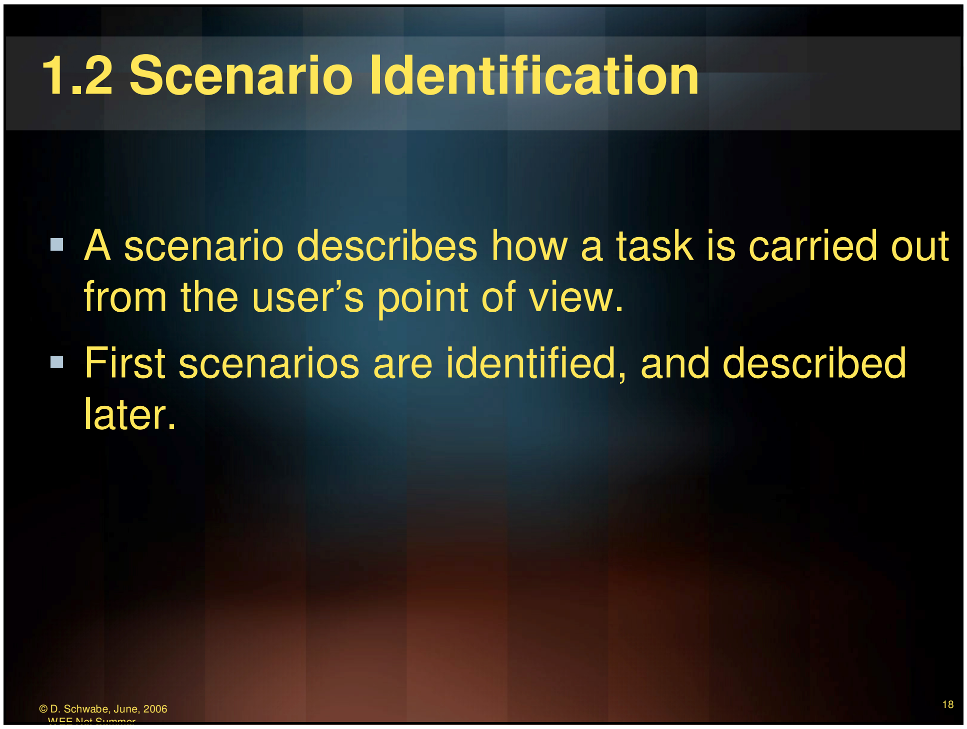
- Analysis of documents and interaction with users themselves.
- Results:

  1. Identification of Classes of Users
  2. Identification of Potential Scenarios



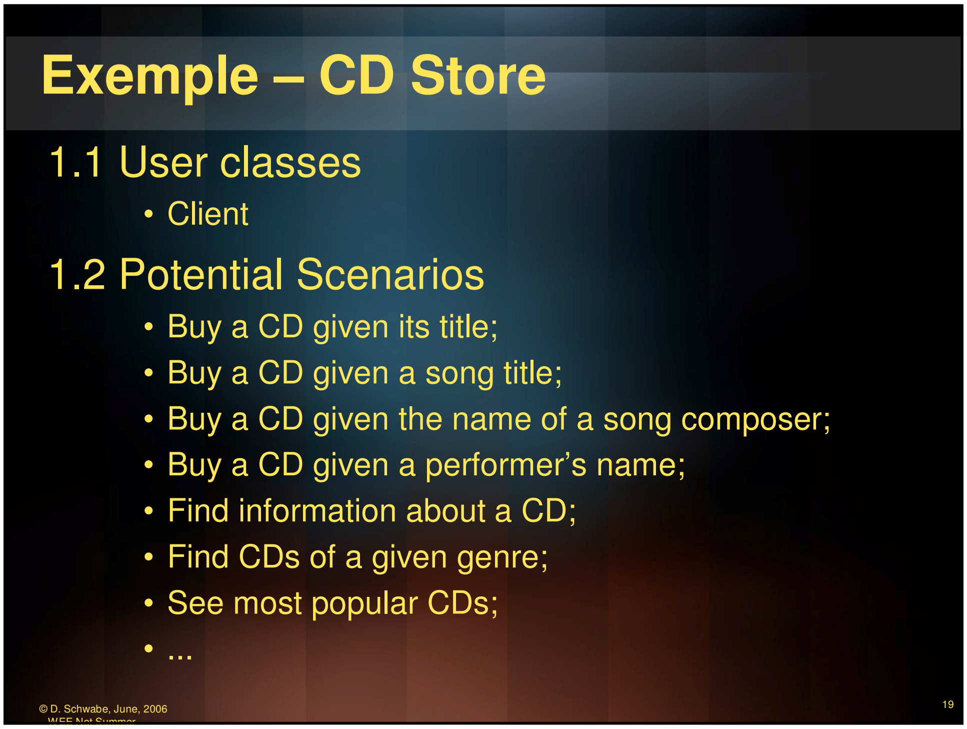
1.1 Identification of Classes of Users

- Users are people that exchange information with the application
- Classes of Users represent users that perform the same set of tasks.
- Note: Users only discuss scenarios for tasks they perform



1.2 Scenario Identification

- A scenario describes how a task is carried out from the user's point of view.
- First scenarios are identified, and described later.



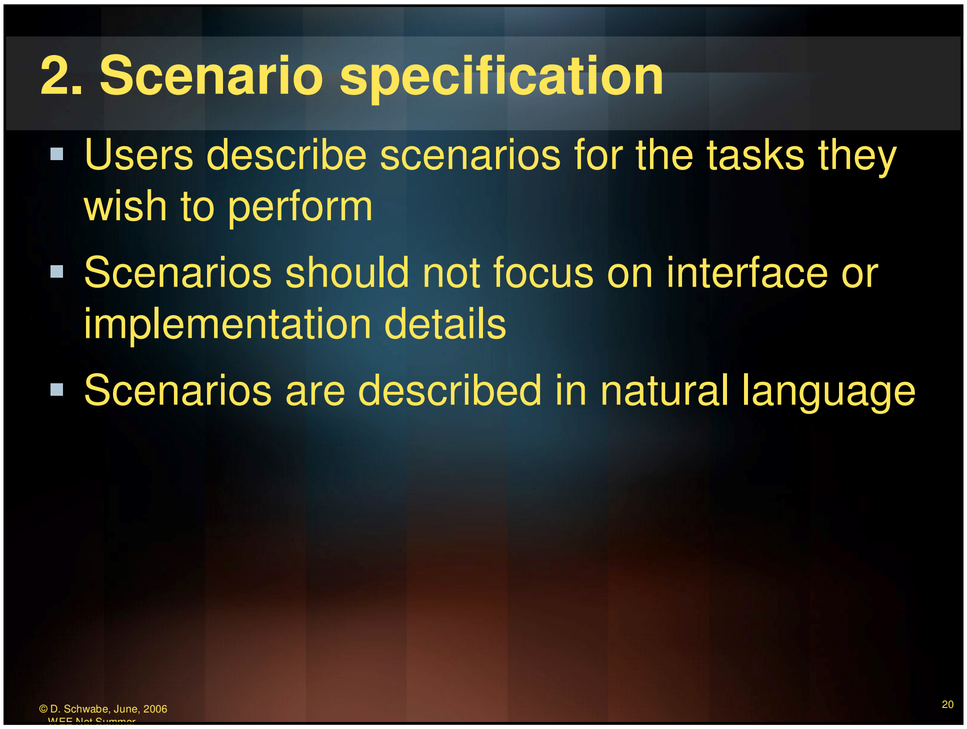
Exemple – CD Store

1.1 User classes

- Client

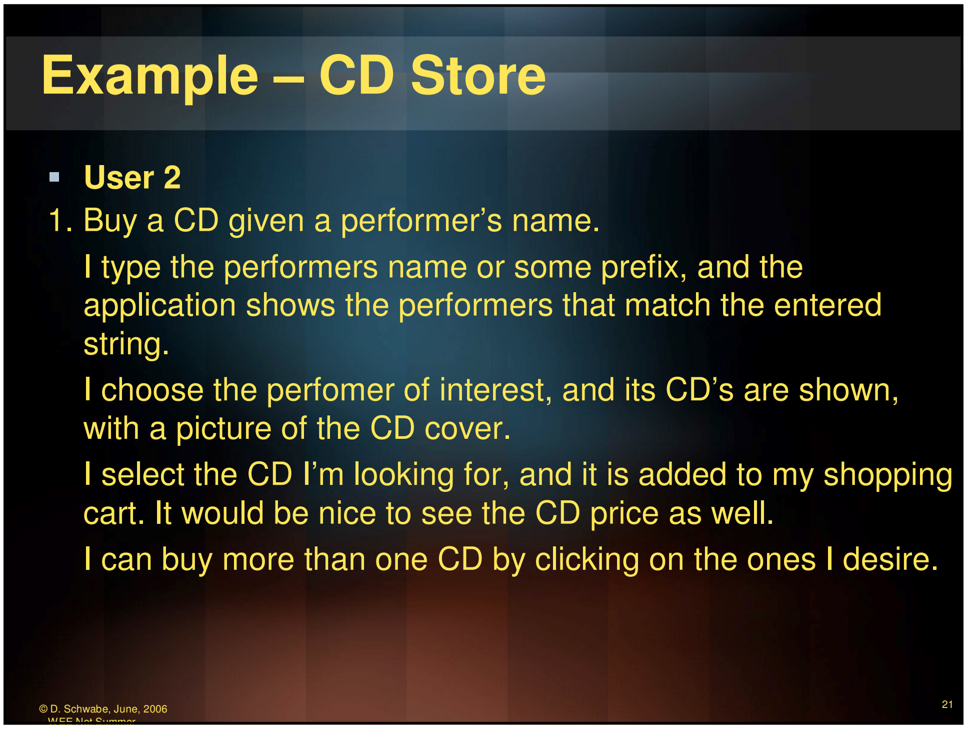
1.2 Potential Scenarios

- Buy a CD given its title;
- Buy a CD given a song title;
- Buy a CD given the name of a song composer;
- Buy a CD given a performer’s name;
- Find information about a CD;
- Find CDs of a given genre;
- See most popular CDs;
- ...



2. Scenario specification

- Users describe scenarios for the tasks they wish to perform
- Scenarios should not focus on interface or implementation details
- Scenarios are described in natural language



Example – CD Store

- **User 2**

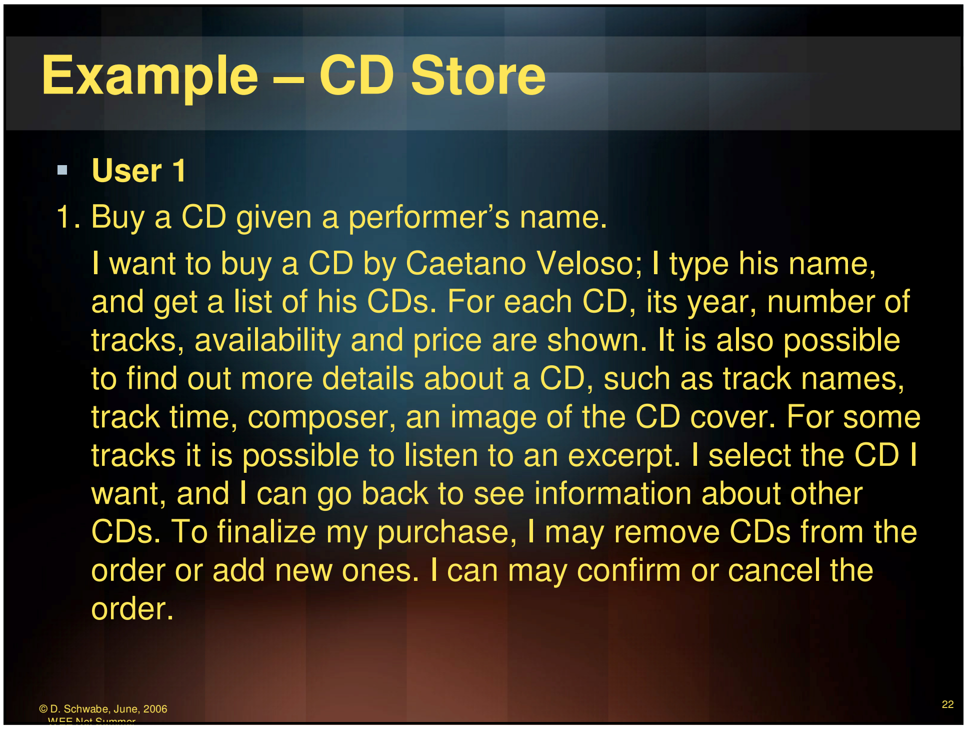
1. Buy a CD given a performer’s name.

   I type the performers name or some prefix, and the application shows the performers that match the entered string.

   I choose the perfomer of interest, and its CD’s are shown, with a picture of the CD cover.

   I select the CD I’m looking for, and it is added to my shopping cart. It would be nice to see the CD price as well.

   I can buy more than one CD by clicking on the ones I desire.

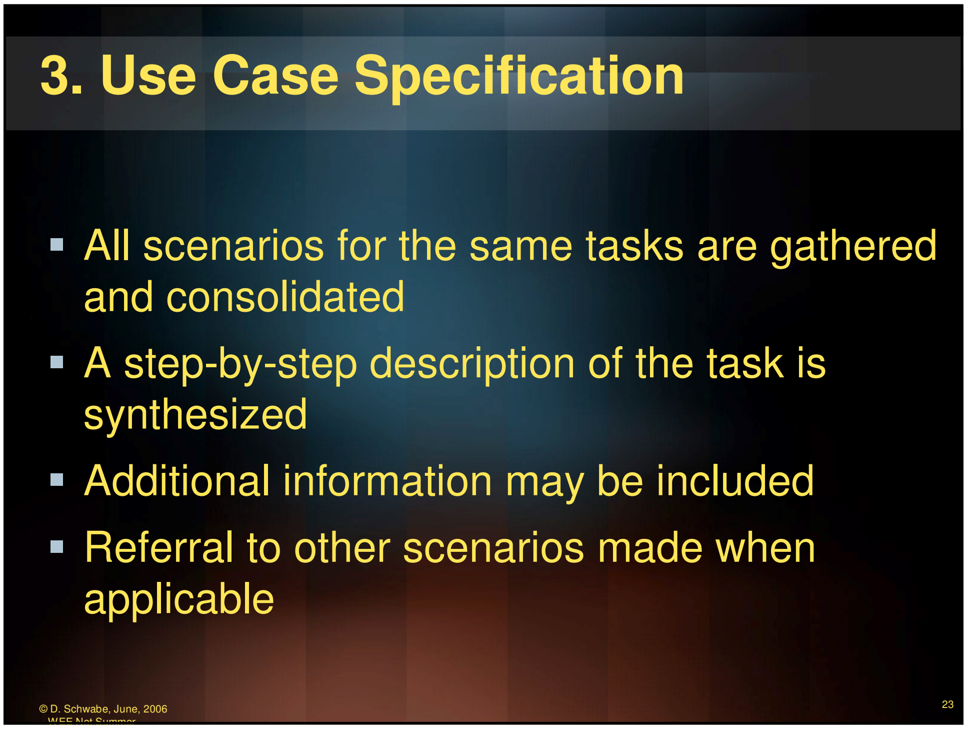


Example – CD Store

- **User 1**

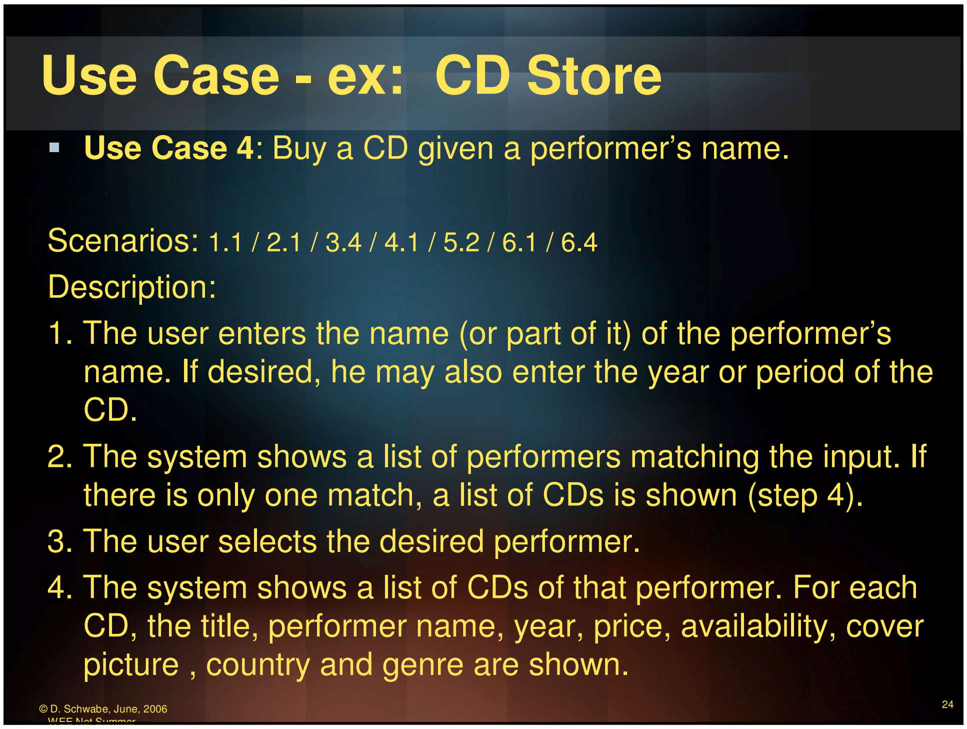
1. Buy a CD given a performer’s name.

   I want to buy a CD by Caetano Veloso; I type his name, and get a list of his CDs. For each CD, its year, number of tracks, availability and price are shown. It is also possible to find out more details about a CD, such as track names, track time, composer, an image of the CD cover. For some tracks it is possible to listen to an excerpt. I select the CD I want, and I can go back to see information about other CDs. To finalize my purchase, I may remove CDs from the order or add new ones. I can may confirm or cancel the order.



3. Use Case Specification

- All scenarios for the same tasks are gathered and consolidated
- A step-by-step description of the task is synthesized
- Additional information may be included
- Referral to other scenarios made when applicable



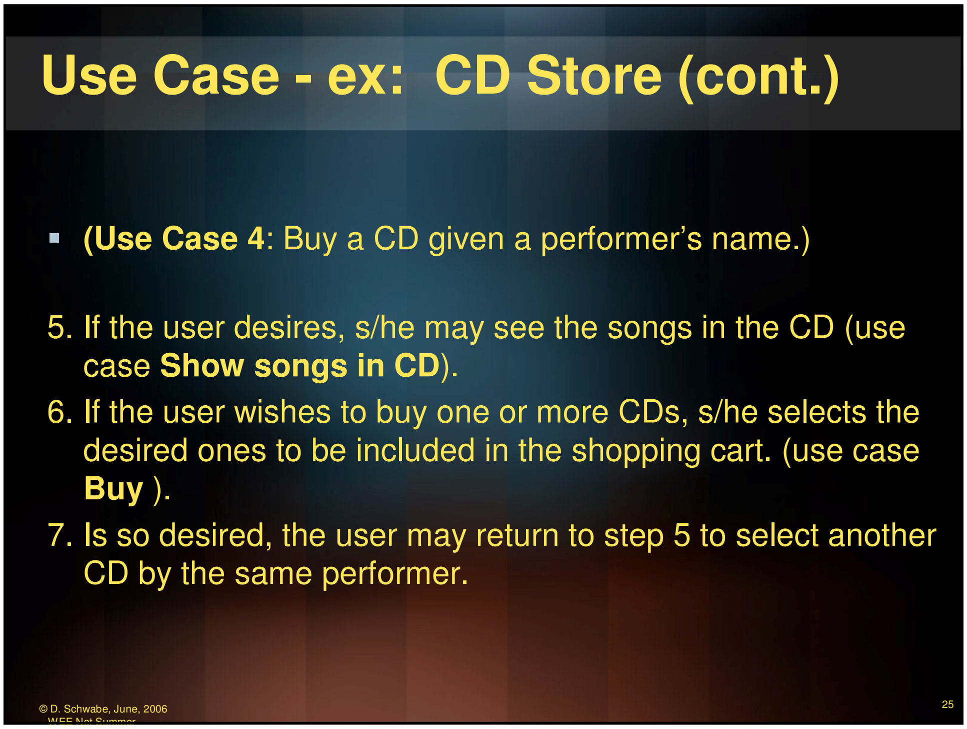
Use Case - ex: CD Store

- **Use Case 4**: Buy a CD given a performer’s name.

Scenarios: 1.1 / 2.1 / 3.4 / 4.1 / 5.2 / 6.1 / 6.4

Description:

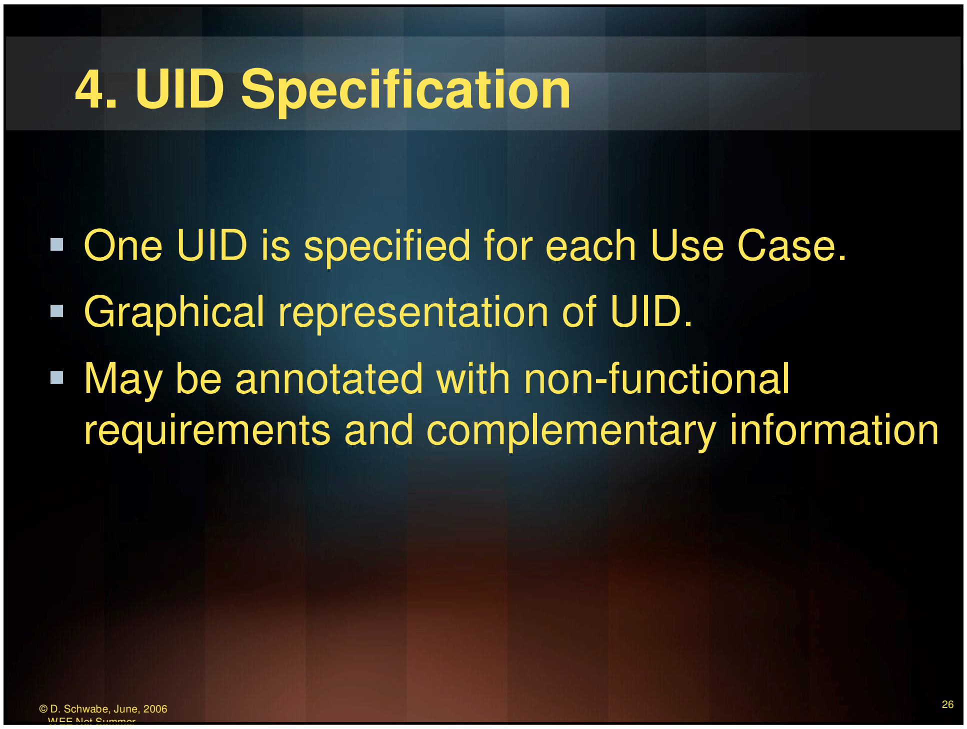
1. The user enters the name (or part of it) of the performer’s name. If desired, he may also enter the year or period of the CD.
2. The system shows a list of performers matching the input. If there is only one match, a list of CDs is shown (step 4).
3. The user selects the desired performer.
4. The system shows a list of CDs of that performer. For each CD, the title, performer name, year, price, availability, cover picture , country and genre are shown.



Use Case - ex: CD Store (cont.)

- **(Use Case 4**: Buy a CD given a performer's name.)

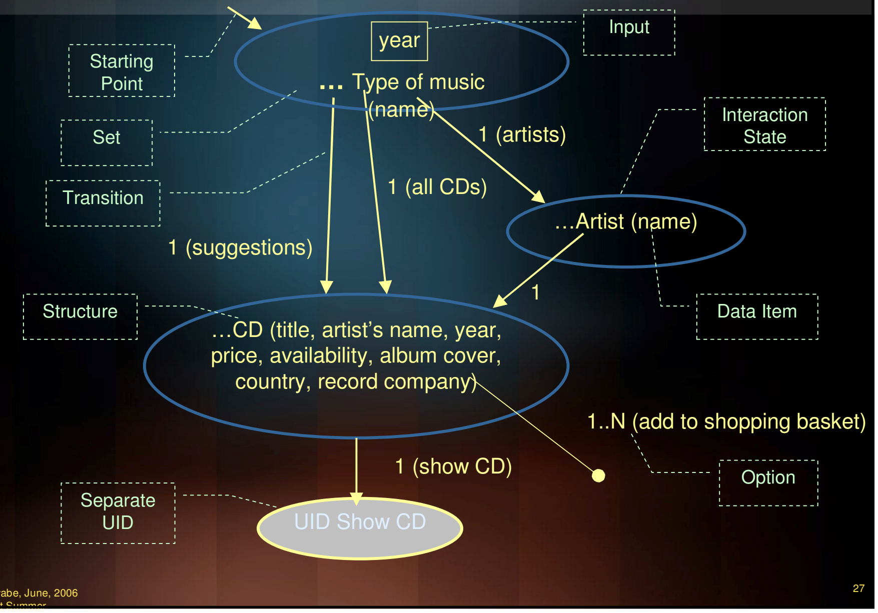
5. If the user desires, s/he may see the songs in the CD (use case **Show songs in CD**).
6. If the user wishes to buy one or more CDs, s/he selects the desired ones to be included in the shopping cart. (use case **Buy**).
7. Is so desired, the user may return to step 5 to select another CD by the same performer.



4. UID Specification

- One UID is specified for each Use Case.
- Graphical representation of UID.
- May be annotated with non-functional requirements and complementary information

UID Notation



year

Input

Starting Point

… Type of music (name)

Set

Interaction State

1 (artists)

Transition

1 (all CDs)

…Artist (name)

1 (suggestions)

Structure

1

Data Item

…CD (title, artist's name, year, price, availability, album cover, country, record company)

1..N (add to shopping basket)

1 (show CD)

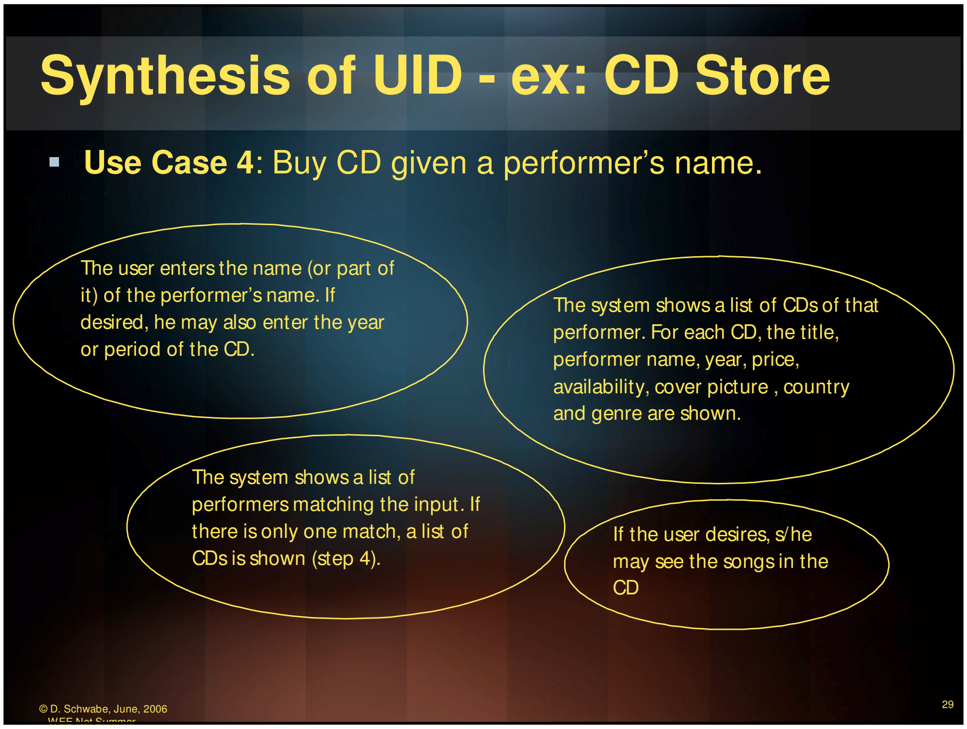
Option

Separate UID

UID Show CD

UIDs - Notation

- Elipsis represent interaction between user and application
- Each elipsis shows the information exchanged between between user and application
- Arrows connecting elipsis represent procssing by the application before new information is exchanged.
- Initial interaction is signaled with sourceless arrow



Synthesis of UID - ex: CD Store

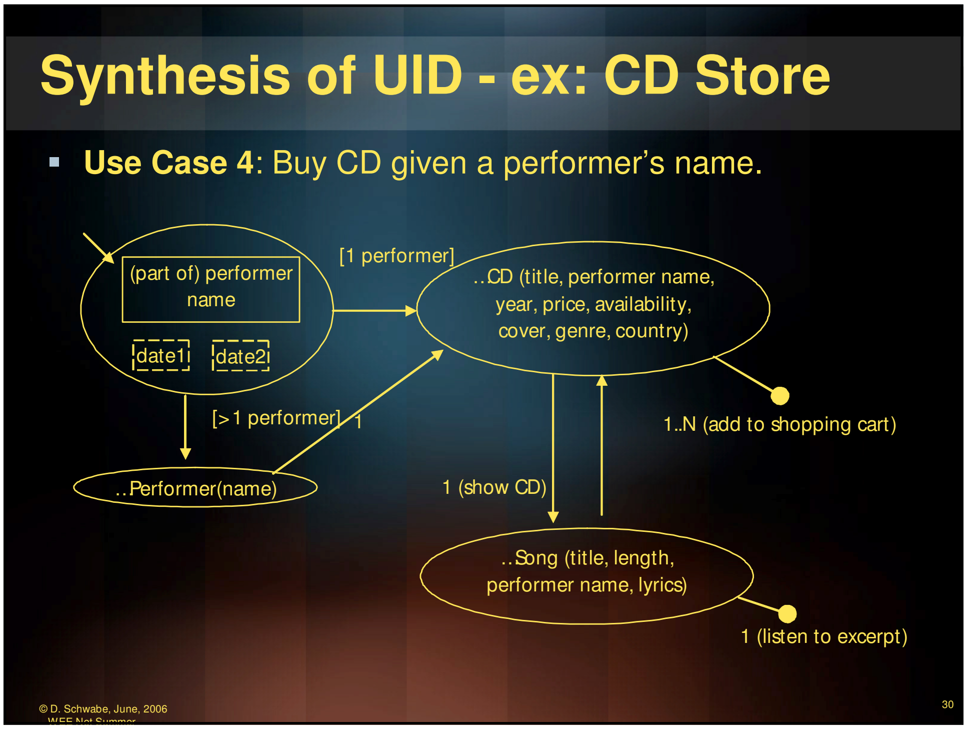
- **Use Case 4**: Buy CD given a performer’s name.

The user enters the name (or part of it) of the performer’s name. If desired, he may also enter the year or period of the CD.

The system shows a list of CDs of that performer. For each CD, the title, performer name, year, price, availability, cover picture , country and genre are shown.

The system shows a list of performers matching the input. If there is only one match, a list of CDs is shown (step 4).

If the user desires, s/he may see the songs in the CD



Synthesis of UID - ex: CD Store

- **Use Case 4**: Buy CD given a performer’s name.

(part of) performer name

date1

date2

[1 performer]

..CD (title, performer name, year, price, availability, cover, genre, country)

[>1 performer]

1

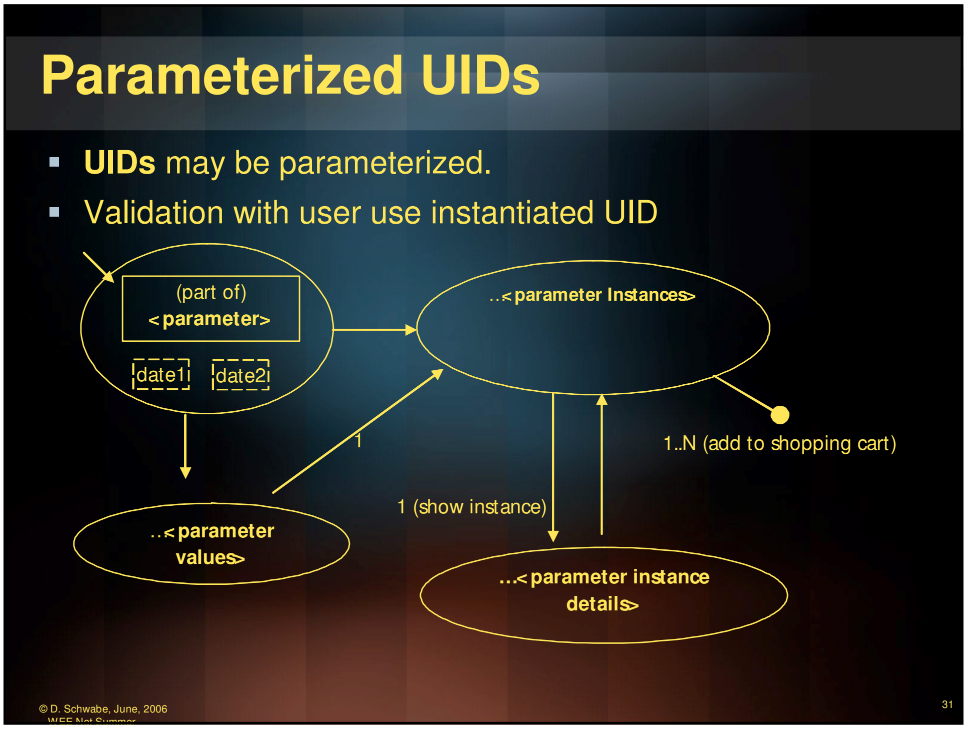
..Performer(name)

1..N (add to shopping cart)

1 (show CD)

..Song (title, length, performer name, lyrics)

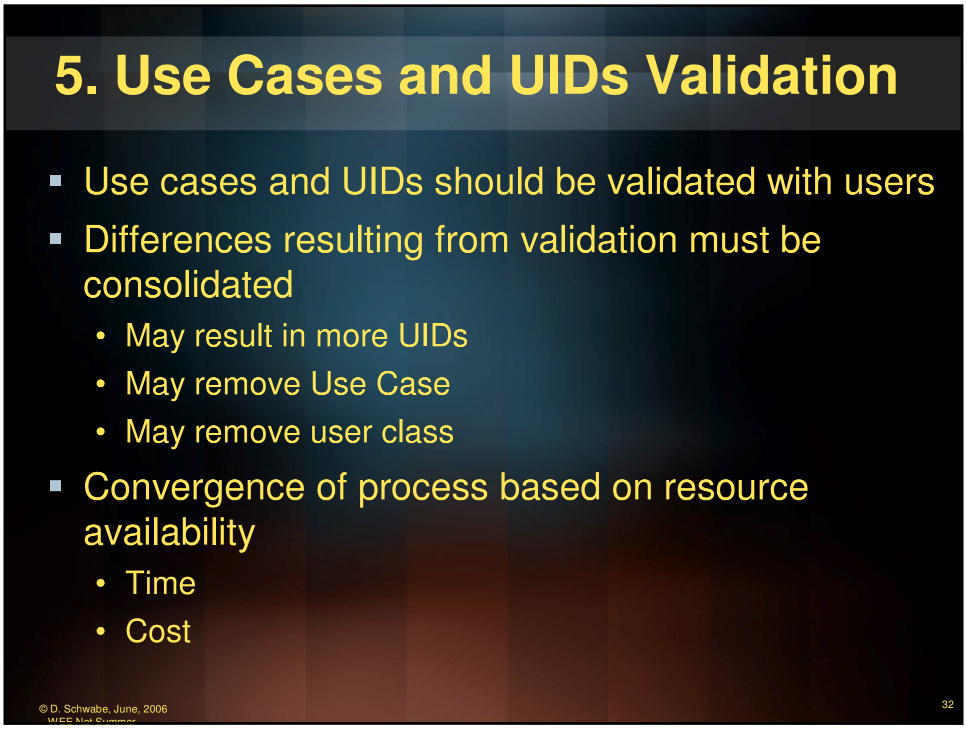
1 (listen to excerpt)



Parameterized UIDs

- **UIDs** may be parameterized.
- Validation with user use instantiated UID

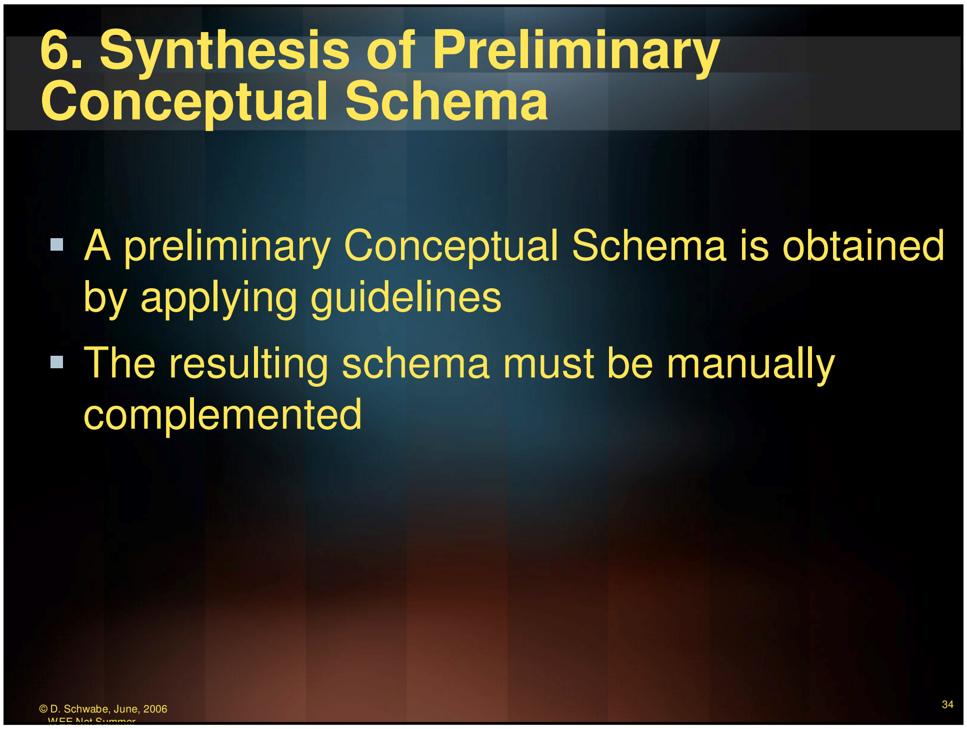
(part of)
<parameter>
date1
date2
..<parameter Instances>
1
1..N (add to shopping cart)
1 (show instance)
..<parameter values>
...<parameter instance details>



5. Use Cases and UIDs Validation

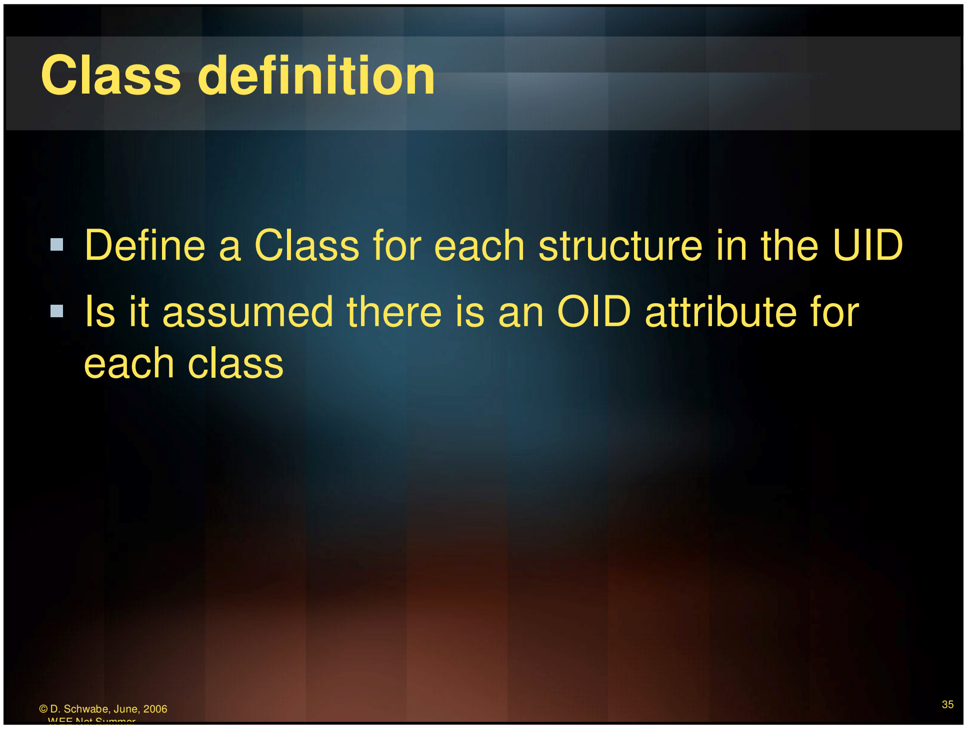
- Use cases and UIDs should be validated with users
- Differences resulting from validation must be consolidated
  - May result in more UIDs
  - May remove Use Case
  - May remove user class
- Convergence of process based on resource availability
  - Time
  - Cost

Conceptual Design



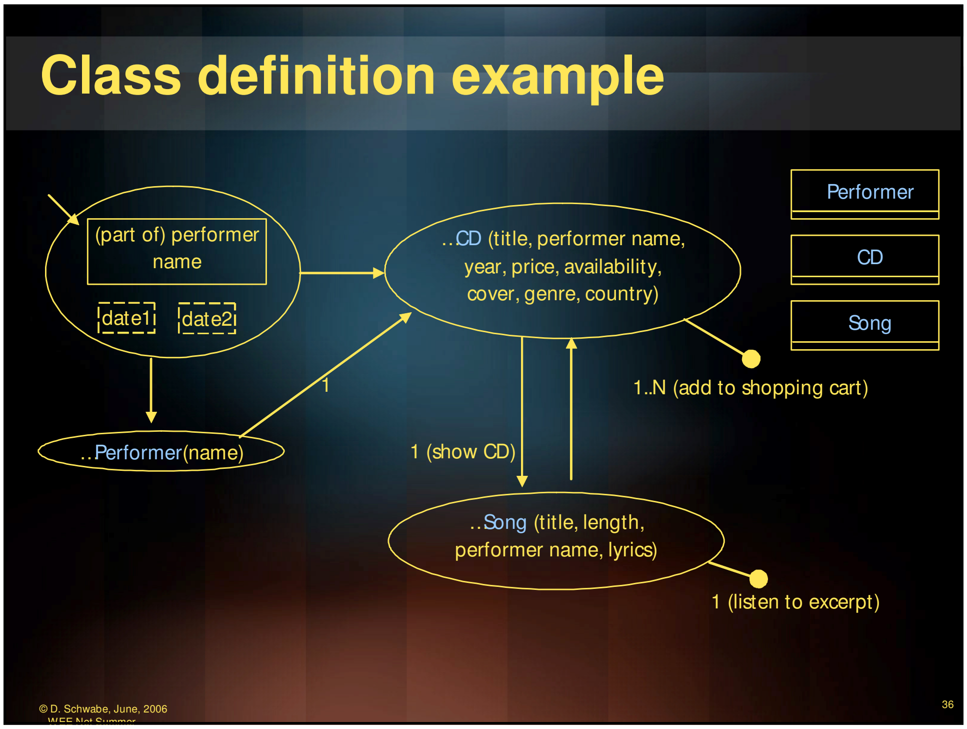
6. Synthesis of Preliminary Conceptual Schema

- A preliminary Conceptual Schema is obtained by applying guidelines
- The resulting schema must be manually complemented



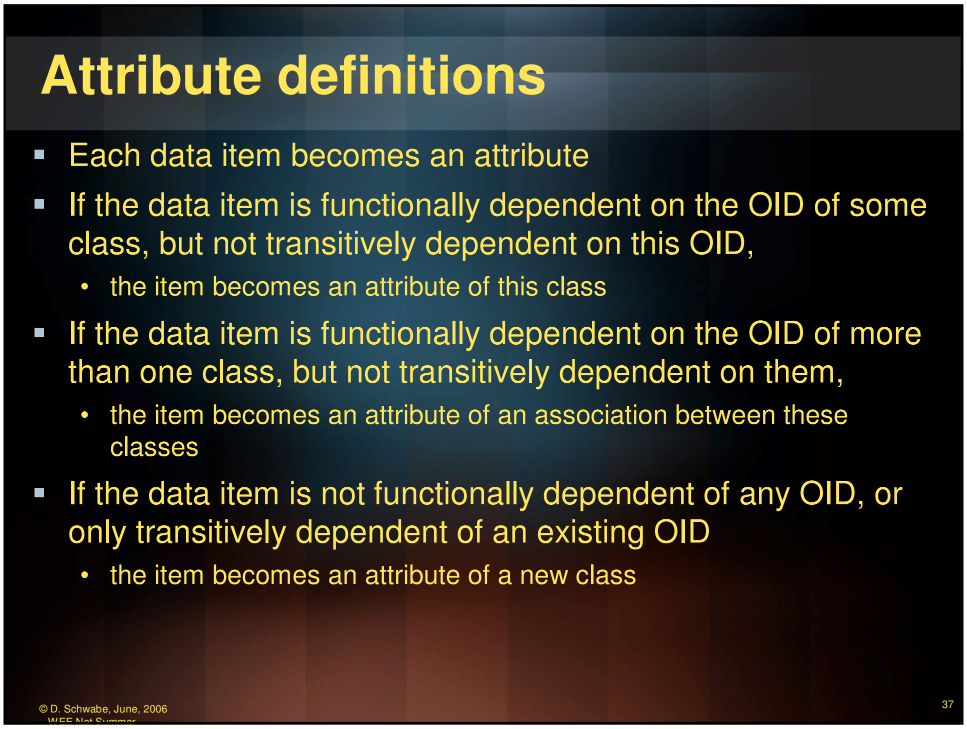
Class definition

- Define a Class for each structure in the UID
- Is it assumed there is an OID attribute for each class



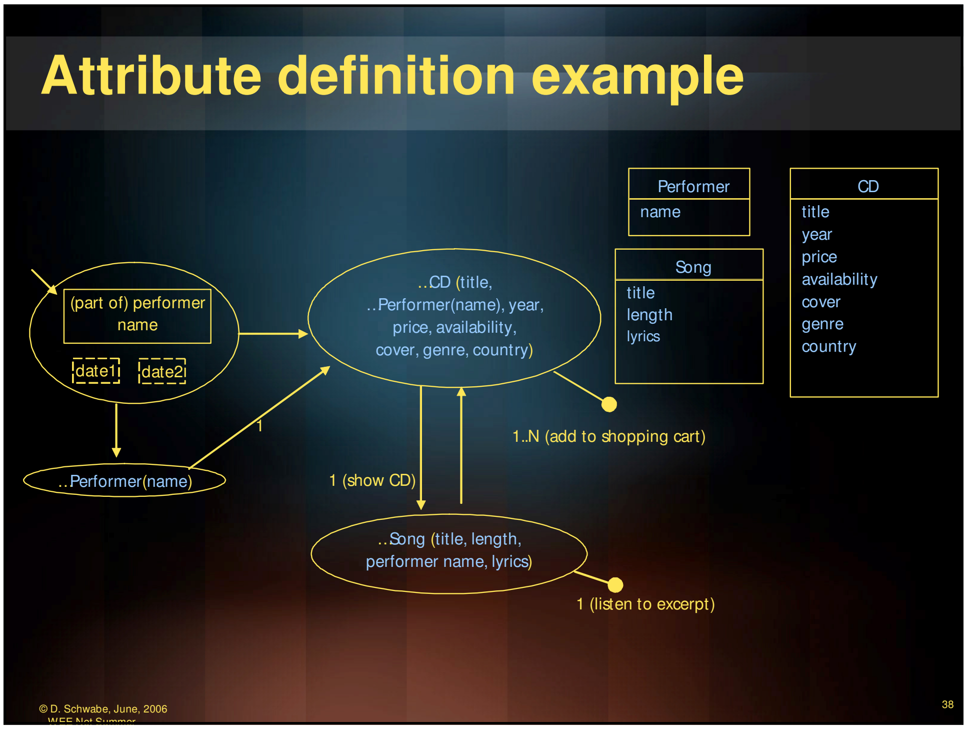
Class definition example

(part of) performer name
date1
date2
..Performer(name)
1
..CD (title, performer name, year, price, availability, cover, genre, country)
1..N (add to shopping cart)
1 (show CD)
..Song (title, length, performer name, lyrics)
1 (listen to excerpt)
Performer
CD
Song

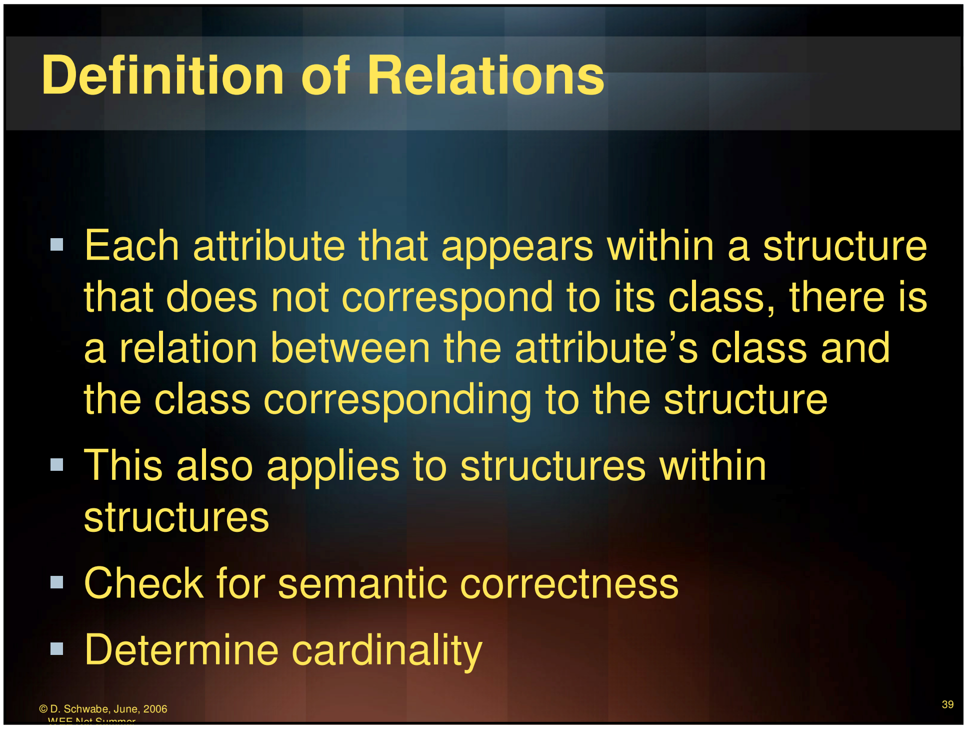


Attribute definitions

- Each data item becomes an attribute
- If the data item is functionally dependent on the OID of some class, but not transitively dependent on this OID,
  - the item becomes an attribute of this class
- If the data item is functionally dependent on the OID of more than one class, but not transitively dependent on them,
  - the item becomes an attribute of an association between these classes
- If the data item is not functionally dependent of any OID, or only transitively dependent of an existing OID
  - the item becomes an attribute of a new class

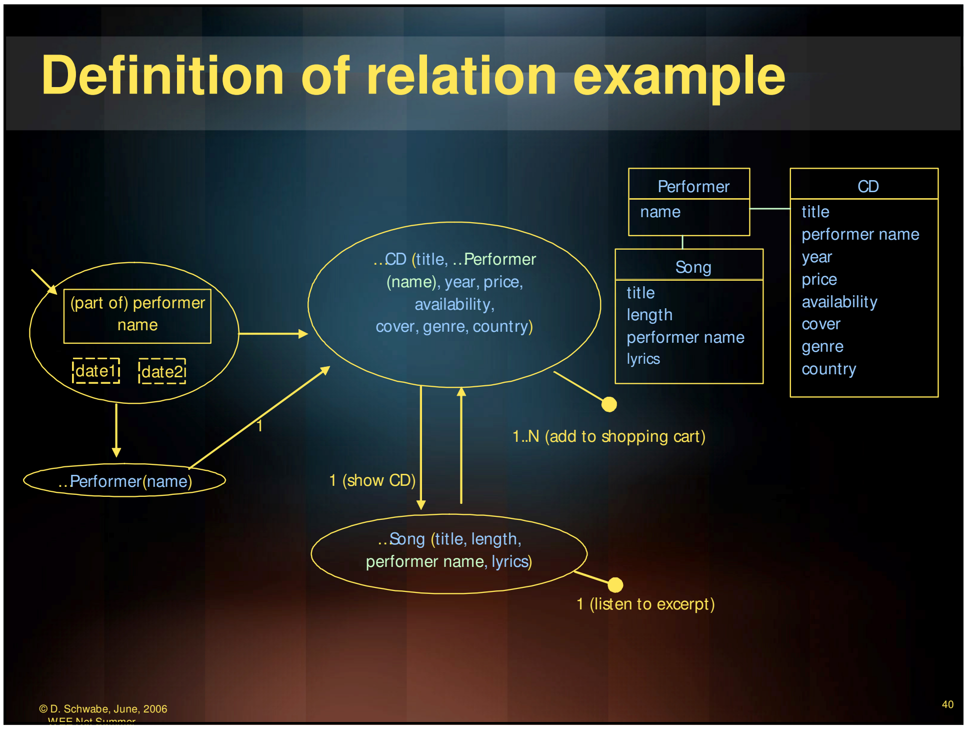


Attribute definition example

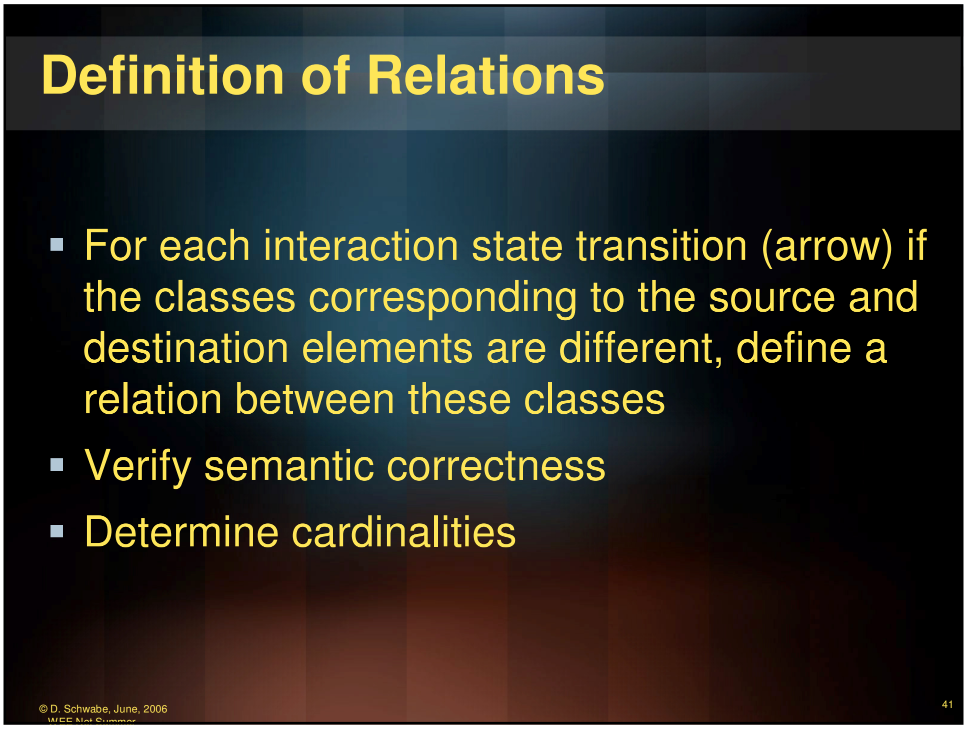


Definition of Relations

- Each attribute that appears within a structure that does not correspond to its class, there is a relation between the attribute’s class and the class corresponding to the structure
- This also applies to structures within structures
- Check for semantic correctness
- Determine cardinality

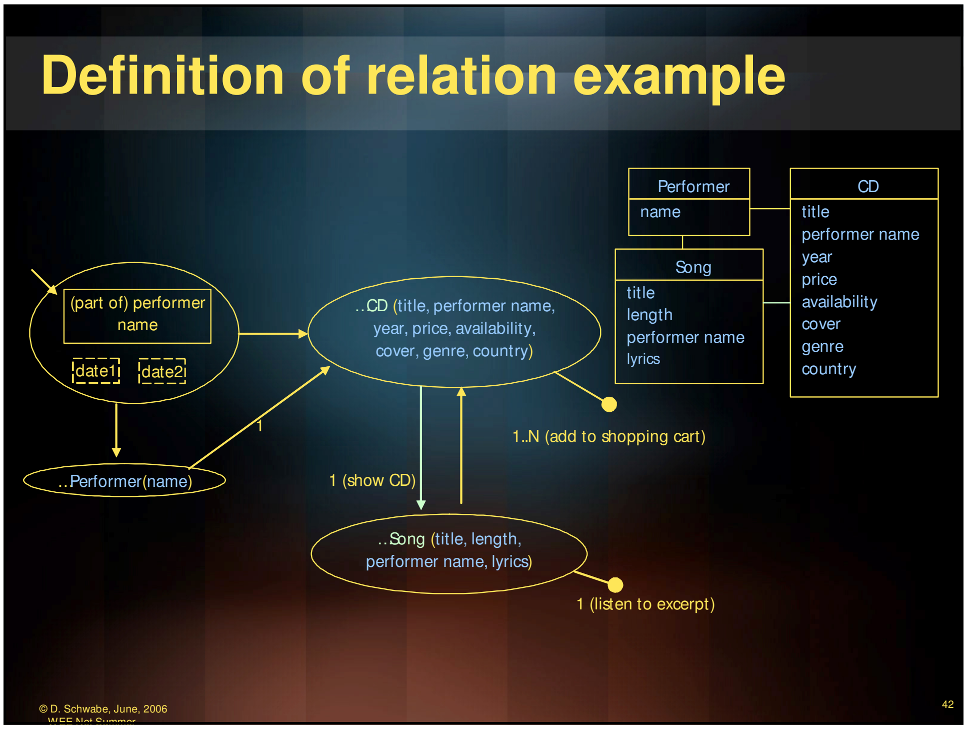


Definition of relation example



Definition of Relations

- For each interaction state transition (arrow) if the classes corresponding to the source and destination elements are different, define a relation between these classes
- Verify semantic correctness
- Determine cardinalities



Definition of relation example

Performer
name

CD
title
performer name
year
price
availability
cover
genre
country

Song
title
length
performer name
lyrics

(part of) performer name
date1
date2

..Performer(name)

1

..CD (title, performer name, year, price, availability, cover, genre, country)

1..N (add to shopping cart)

1 (show CD)

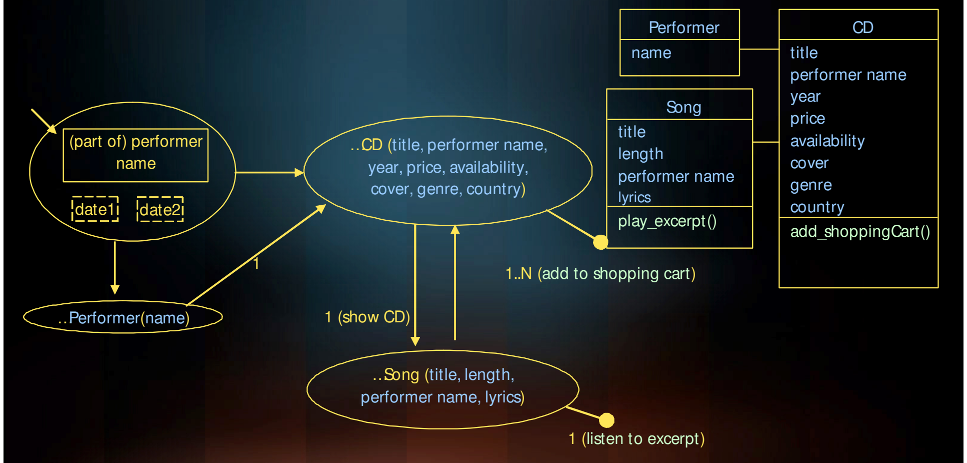
..Song (title, length, performer name, lyrics)

1 (listen to excerpt)

Definition of Operations

- For each option leading to an empty state, define an operation for the associated class
- Verify semantic correctness

Definition of relation example



Performer
name

CD
title
performer name
year
price
availability
cover
genre
country
add_shoppingCart()

Song
title
length
performer name
lyrics
play_excerpt()

(part of) performer name

date1

date2

..Performer(name)

1

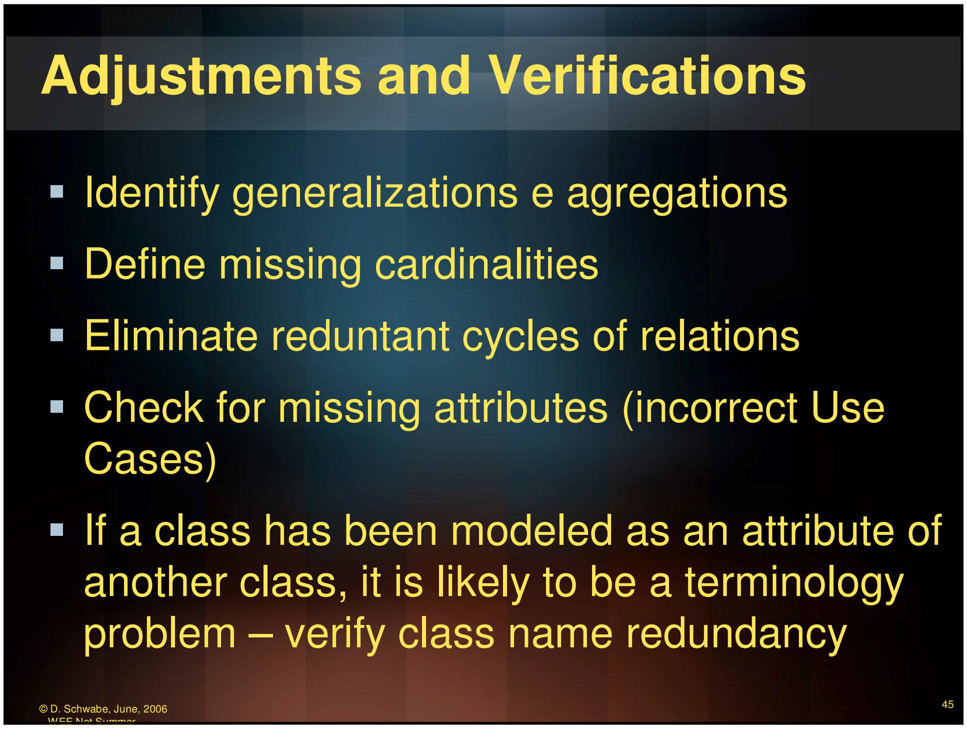
..CD (title, performer name, year, price, availability, cover, genre, country)

1..N (add to shopping cart)

1 (show CD)

..Song (title, length, performer name, lyrics)

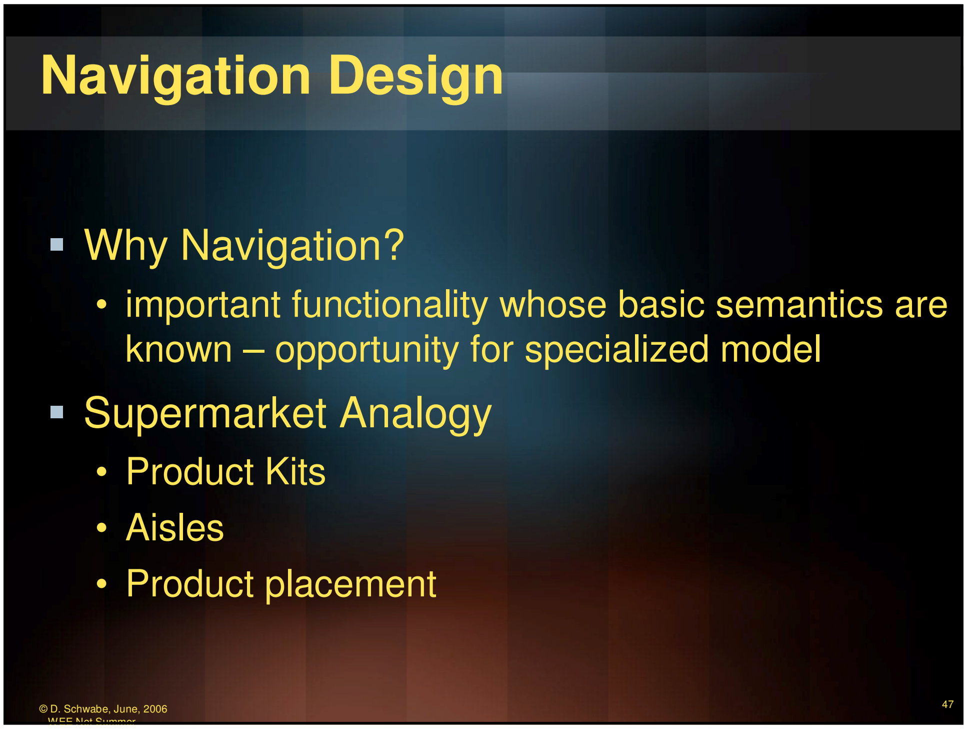
1 (listen to excerpt)



Adjustments and Verifications

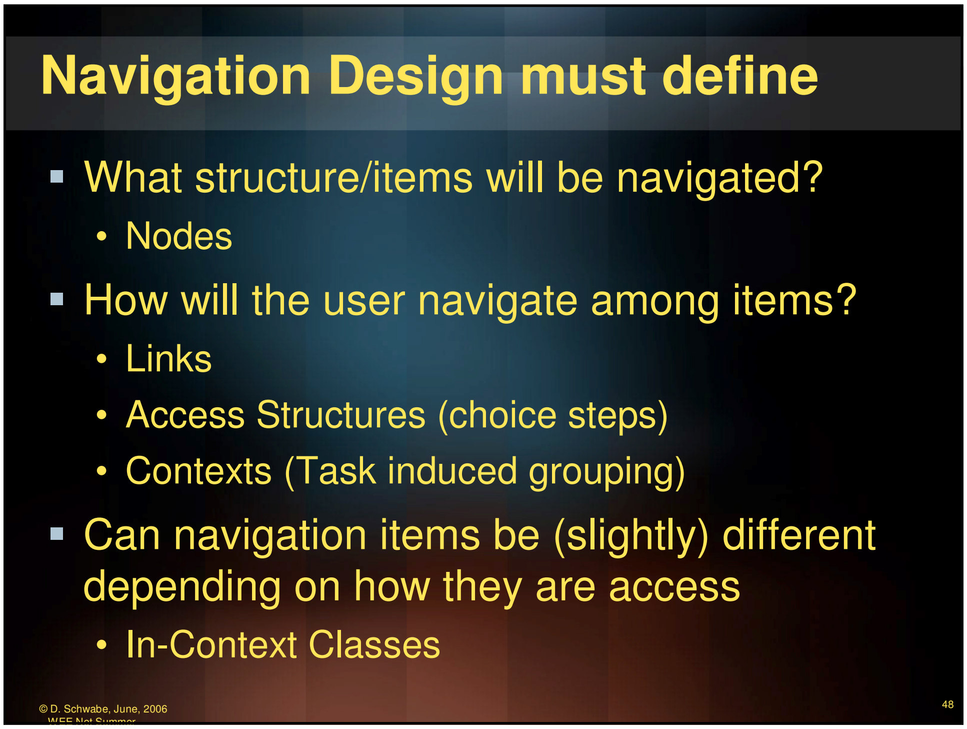
- Identify generalizations e agregations
- Define missing cardinalities
- Eliminate reduntant cycles of relations
- Check for missing attributes (incorrect Use Cases)
- If a class has been modeled as an attribute of another class, it is likely to be a terminology problem – verify class name redundancy

Navigation Design



Navigation Design

- Why Navigation?
  - important functionality whose basic semantics are known – opportunity for specialized model
- Supermarket Analogy
  - Product Kits
  - Aisles
  - Product placement



Navigation Design must define

- What structure/items will be navigated?
  - Nodes
- How will the user navigate among items?
  - Links
  - Access Structures (choice steps)
  - Contexts (Task induced grouping)
- Can navigation items be (slightly) different depending on how they are access
  - In-Context Classes

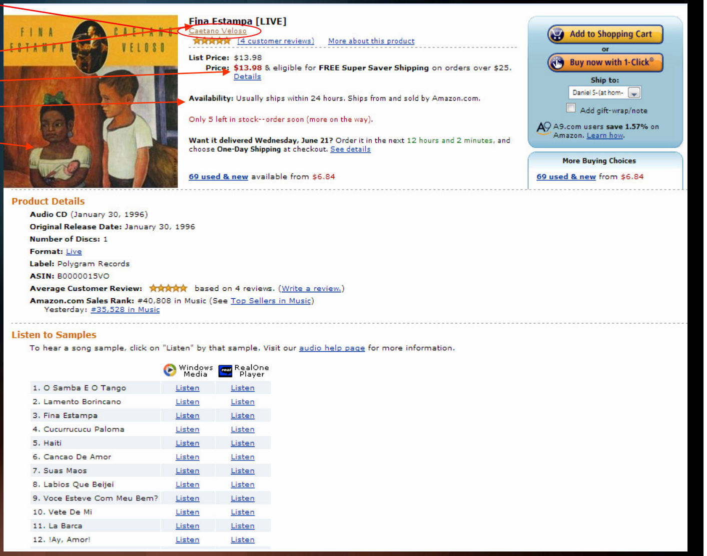
Navigation Object

?

CD

title
price
availability
cover

add_shoppingCart()



Fina Estampa [LIVE]
Caetano Veloso
★★★★★ (4 customer reviews) More about this product

List Price: $13.98
Price: $13.98 & eligible for **FREE Super Saver Shipping** on orders over $25. Details

Availability: Usually ships within 24 hours. Ships from and sold by Amazon.com.

Only 5 left in stock--order soon (more on the way).

Want it delivered Wednesday, June 21? Order it in the next 12 hours and 2 minutes, and choose **One-Day Shipping** at checkout. See details

69 used & new available from $6.84

Ship to: Daniel S-(at hom-
□ Add gift-wrap/note
A9.com users **save 1.57%** on Amazon. Learn how.

More Buying Choices
69 used & new from $6.84

Product Details

Audio CD (January 30, 1996)
Original Release Date: January 30, 1996
Number of Discs: 1
Format: Live
Label: Polygram Records
ASIN: B0000015VO
Average Customer Review: ★★★★★ based on 4 reviews. (Write a review.)
Amazon.com Sales Rank: #40,808 in Music (See Top Sellers in Music)
Yesterday: #35,528 in Music

Listen to Samples

To hear a song sample, click on "Listen" by that sample. Visit our audio help page for more information.

	Windows Media	RealOne Player
1. O Samba E O Tango	Listen	Listen
2. Lamento Borincano	Listen	Listen
3. Fina Estampa	Listen	Listen
4. Cucurrucucu Paloma	Listen	Listen
5. Haiti	Listen	Listen
6. Cancao De Amor	Listen	Listen
7. Suas Maos	Listen	Listen
8. Labios Que Beijei	Listen	Listen
9. Voce Esteve Com Meu Bem?	Listen	Listen
10. Vete De Mi	Listen	Listen
11. La Barca	Listen	Listen
12. !Ay, Amor!	Listen	Listen

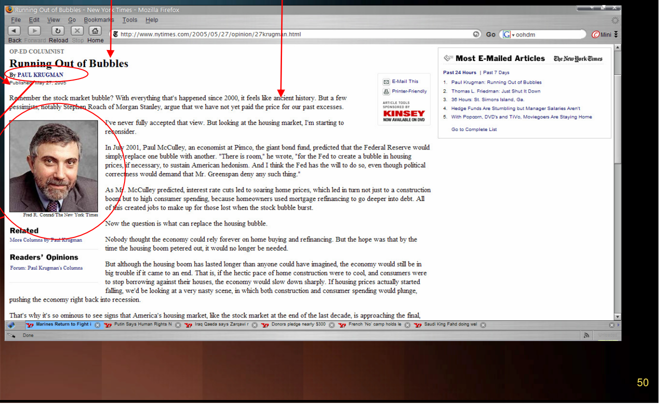
- What are the navigation objects?

Navigation Objects

NewsStory

title: string
content: text
photo: image*
publicationDate: date

- What are the navigation objects?



Running Out of Bubbles - New York Times - Mozilla Firefox

http://www.nytimes.com/2005/05/27/opinion/27krugman.html

OP-ED COLUMNIST

Running Out of Bubbles

By PAUL KRUGMAN
Published: May 27, 2005

Remember the stock market bubble? With everything that's happened since 2000, it feels like ancient history. But a few pessimists, notably Stephen Roach of Morgan Stanley, argue that we have not yet paid the price for our past excesses.

I've never fully accepted that view. But looking at the housing market, I'm starting to reconsider.

In July 2001, Paul McCulley, an economist at Pimco, the giant bond fund, predicted that the Federal Reserve would simply replace one bubble with another. "There is room," he wrote, "for the Fed to create a bubble in housing prices, if necessary, to sustain American hedonism. And I think the Fed has the will to do so, even though political correctness would demand that Mr. Greenspan deny any such thing."

As Mr. McCulley predicted, interest rate cuts led to soaring home prices, which led in turn not just to a construction boom but to high consumer spending, because homeowners used mortgage refinancing to go deeper into debt. All of this created jobs to make up for those lost when the stock bubble burst.

Now the question is what can replace the housing bubble.

Nobody thought the economy could rely forever on home buying and refinancing. But the hope was that by the time the housing boom petered out, it would no longer be needed.

But although the housing boom has lasted longer than anyone could have imagined, the economy would still be in big trouble if it came to an end. That is, if the hectic pace of home construction were to cool, and consumers were to stop borrowing against their houses, the economy would slow down sharply. If housing prices actually started falling, we'd be looking at a very nasty scene, in which both construction and consumer spending would plunge, pushing the economy right back into recession.

That's why it's so ominous to see signs that America's housing market, like the stock market at the end of the last decade, is approaching the final,

Fred R. Conrad/The New York Times

Related
More Columns by Paul Krugman

Readers' Opinions
Forum: Paul Krugman's Columns

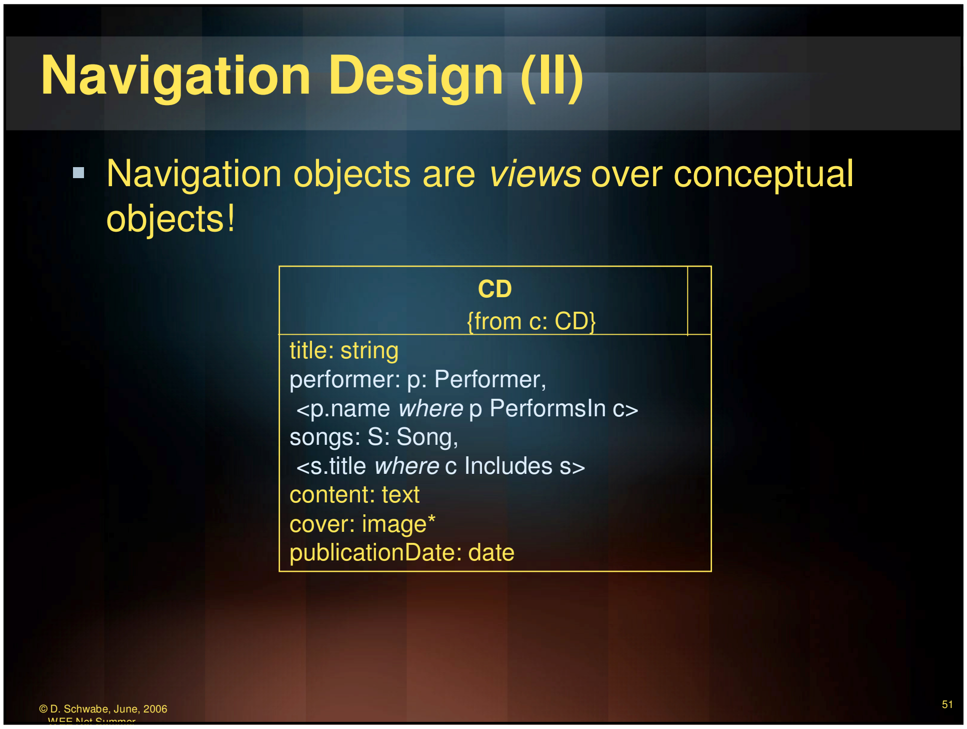
E-Mail This
Printer-Friendly
ARTICLE TOOLS SPONSORED BY KINSEY NOW AVAILABLE ON DVD

Most E-Mailed Articles The New York Times
Past 24 Hours | Past 7 Days
1. Paul Krugman: Running Out of Bubbles
2. Thomas L. Friedman: Just Shut It Down
3. 36 Hours: St. Simons Island, Ga.
4. Hedge Funds Are Stumbling but Manager Salaries Aren't
5. With Popcorn, DVD's and TiVo, Moviegoers Are Staying Home
Go to Complete List

Marines Return to Fight i | Putin Says Human Rights N | Iraq Qaeda says Zarqawi r | Donors pledge nearly $300 | French 'No' camp holds le | Saudi King Fahd doing wel

Done

?

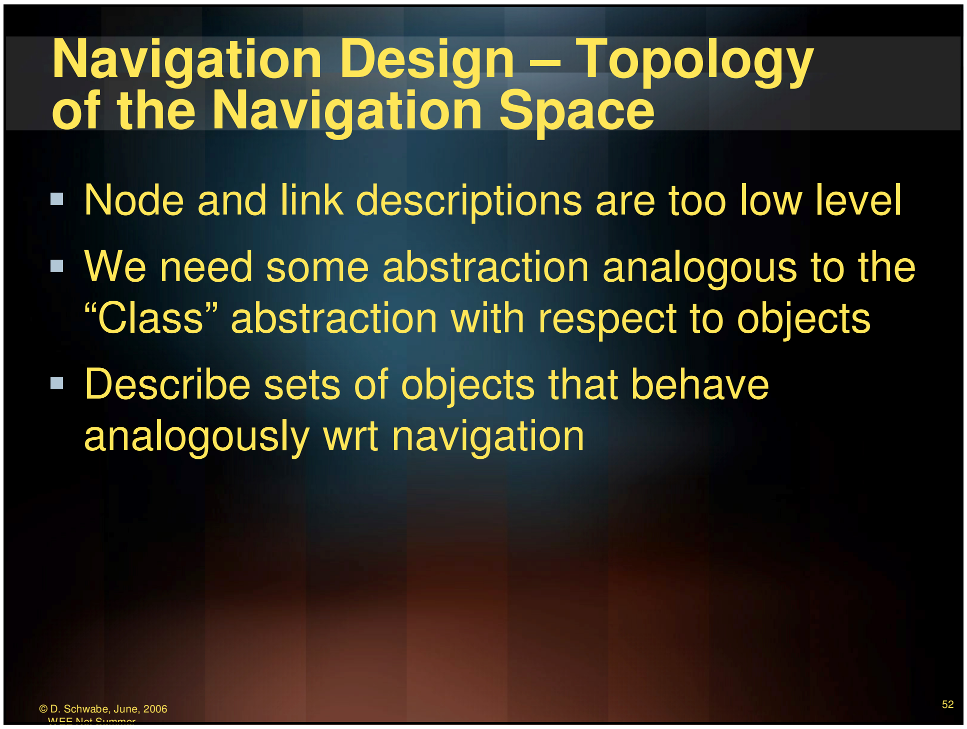


Navigation Design (II)

- Navigation objects are *views* over conceptual objects!

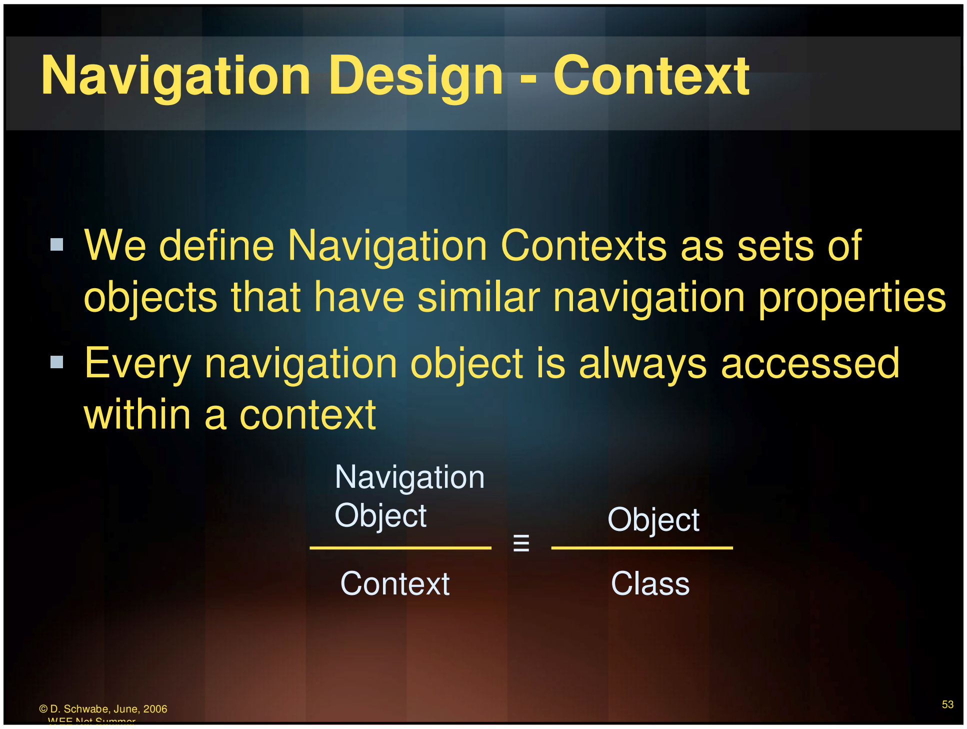
CD
{from c: CD}

title: string
performer: p: Performer,
 <p.name *where* p PerformsIn c>
songs: S: Song,
 <s.title *where* c Includes s>
content: text
cover: image*
publicationDate: date



Navigation Design – Topology of the Navigation Space

- Node and link descriptions are too low level
- We need some abstraction analogous to the "Class" abstraction with respect to objects
- Describe sets of objects that behave analogously wrt navigation

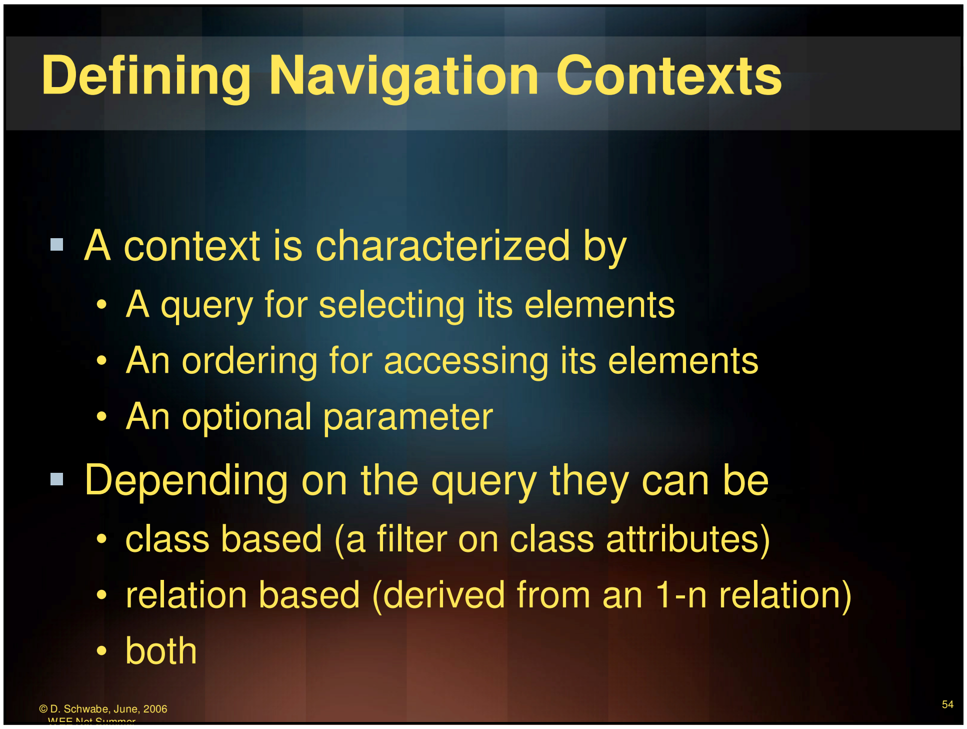


Navigation Design - Context

- We define Navigation Contexts as sets of objects that have similar navigation properties
- Every navigation object is always accessed within a context

$$\frac{\text{Navigation Object}}{\text{Context}} \equiv \frac{\text{Object}}{\text{Class}}$$

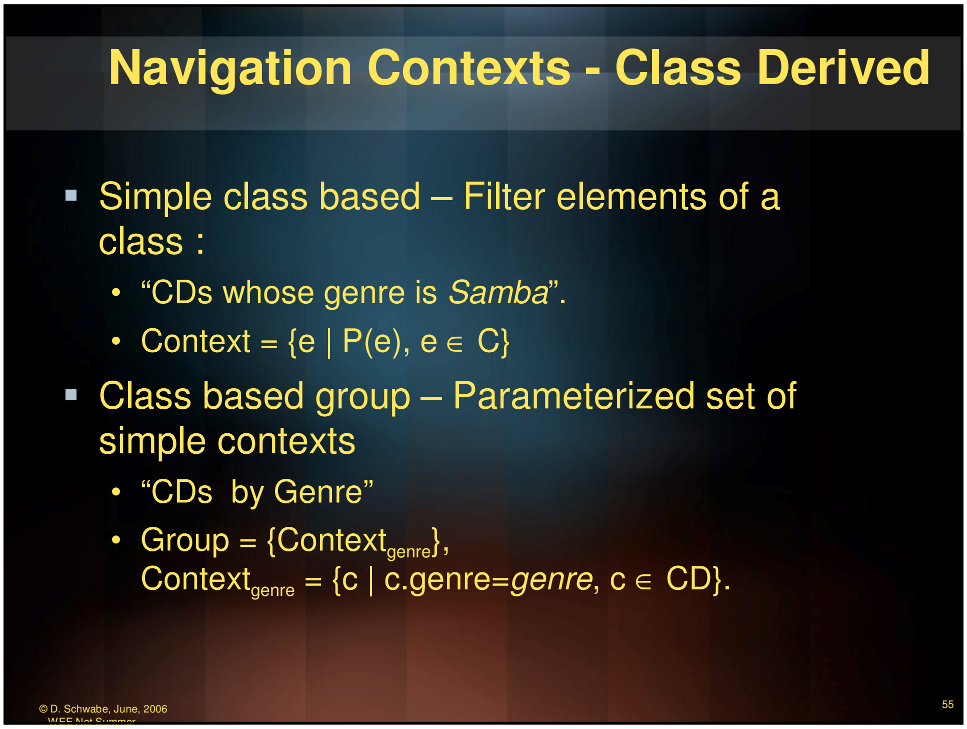
© D. Schwabe, June, 2006
WEE Net Summer



Defining Navigation Contexts

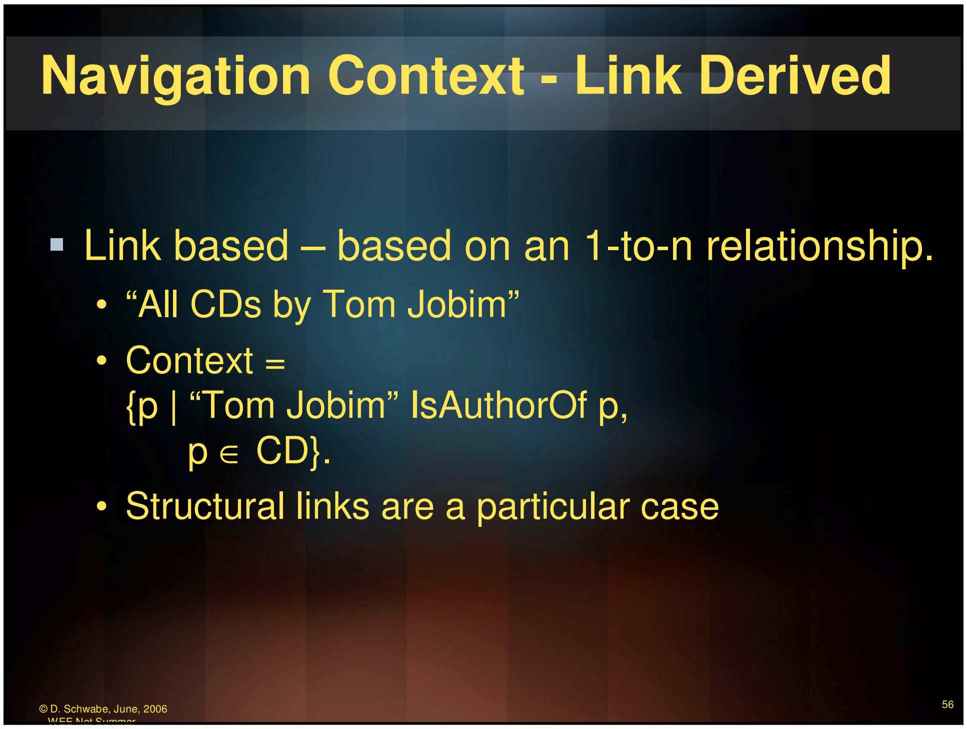
- A context is characterized by
  - A query for selecting its elements
  - An ordering for accessing its elements
  - An optional parameter
- Depending on the query they can be
  - class based (a filter on class attributes)
  - relation based (derived from an 1-n relation)
  - both

© D. Schwabe, June, 2006
WEE Net Summer



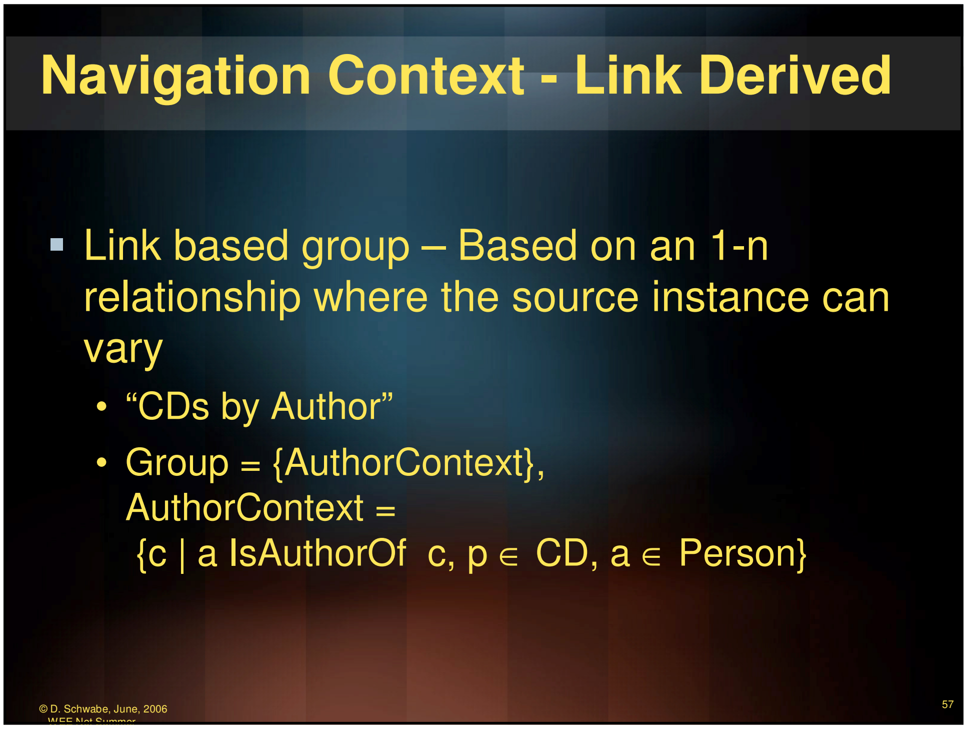
Navigation Contexts - Class Derived

- Simple class based – Filter elements of a class :
  - “CDs whose genre is *Samba*”.
  - Context = $\{e \mid P(e), e \in C\}$
- Class based group – Parameterized set of simple contexts
  - “CDs by Genre”
  - Group = $\{Context_{genre}\}$,
Context$_{genre}$ = $\{c \mid c.genre = \mathit{genre}, c \in CD\}$.



Navigation Context - Link Derived

- Link based – based on an 1-to-n relationship.
  - “All CDs by Tom Jobim”
  - Context =
    {p | “Tom Jobim” IsAuthorOf p,
    p ∈ CD}.
  - Structural links are a particular case

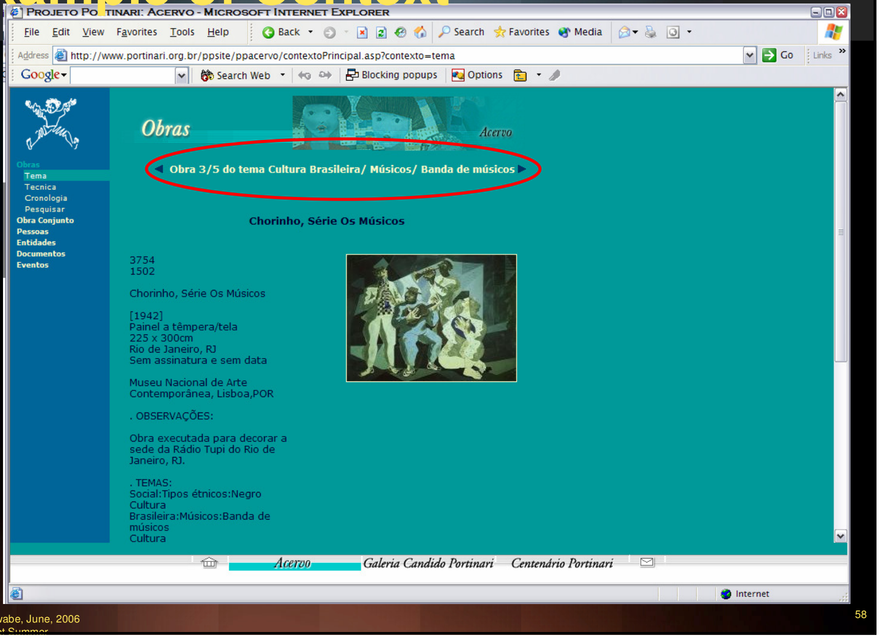


Navigation Context - Link Derived

- Link based group – Based on an 1-n relationship where the source instance can vary
  - “CDs by Author”
  - Group = {AuthorContext}, AuthorContext = {c | a IsAuthorOf c, p ∊ CD, a ∊ Person}

© D. Schwabe, June, 2006

Example of Context



http://www.portinari.org.br/ppsite/ppacervo/contextoPrincipal.asp?contexto=tema

◀ Obra 3/5 do tema Cultura Brasileira/ Músicos/ Banda de músicos ▶

Chorinho, Série Os Músicos

3754
1502

Chorinho, Série Os Músicos

[1942]
Painel a têmpera/tela
225 x 300cm
Rio de Janeiro, RJ
Sem assinatura e sem data

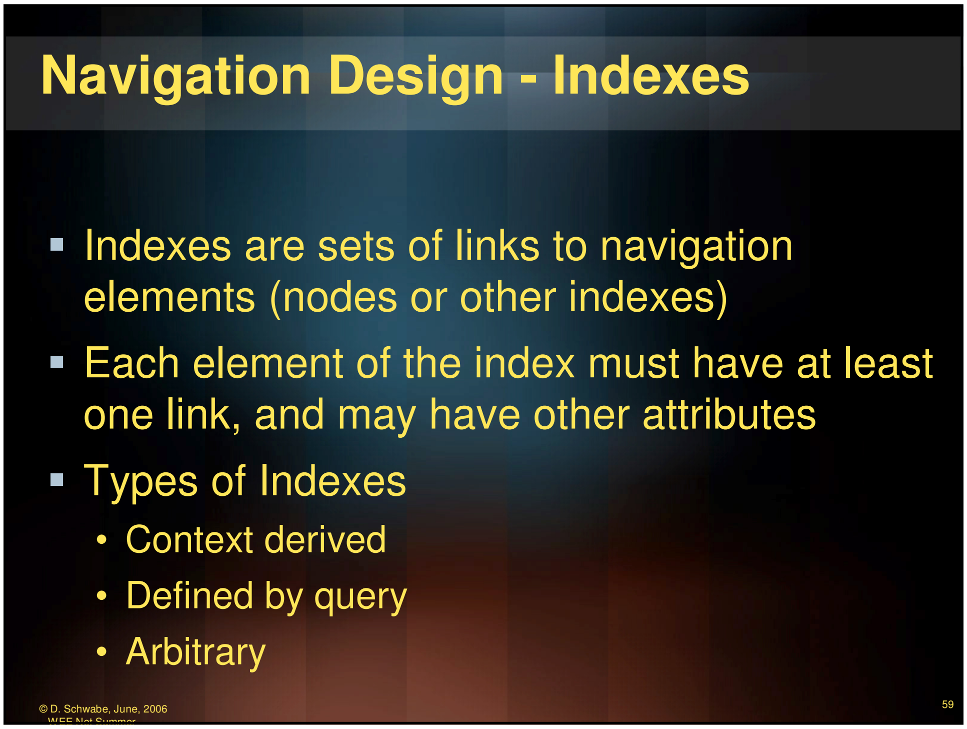
Museu Nacional de Arte
Contemporânea, Lisboa,POR

. OBSERVAÇÕES:

Obra executada para decorar a sede da Rádio Tupi do Rio de Janeiro, RJ.

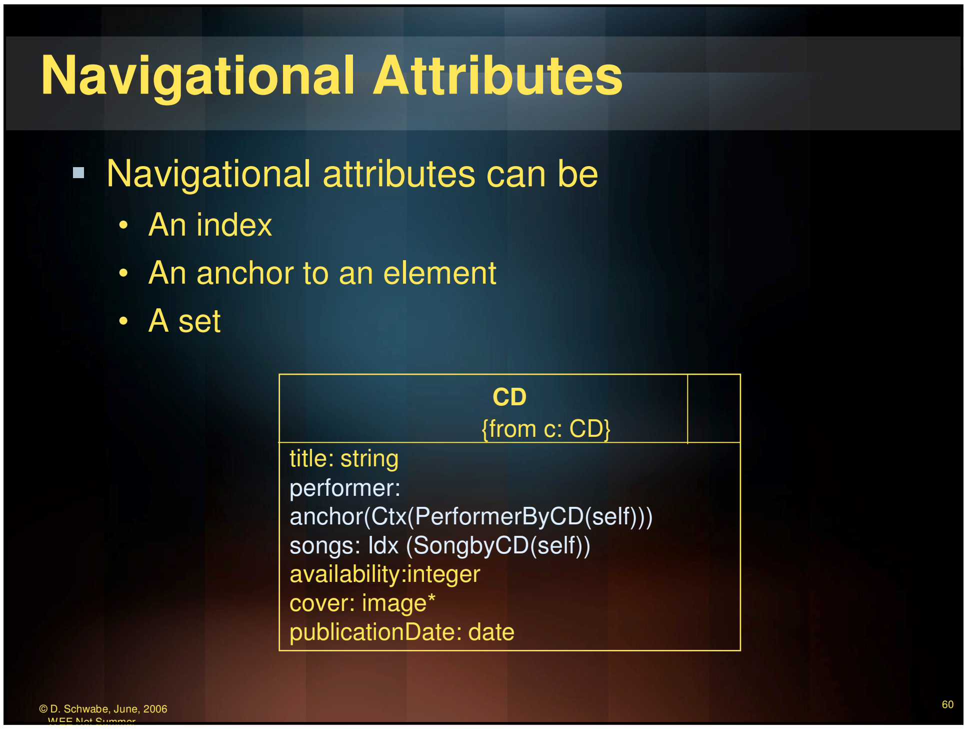
. TEMAS:
Social:Tipos étnicos:Negro
Cultura Brasileira:Músicos:Banda de músicos
Cultura

© D. Schwabe, June, 2006



Navigation Design - Indexes

- Indexes are sets of links to navigation elements (nodes or other indexes)
- Each element of the index must have at least one link, and may have other attributes
- Types of Indexes
  - Context derived
  - Defined by query
  - Arbitrary



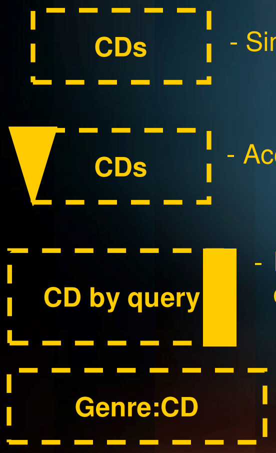
Navigational Attributes

- Navigational attributes can be
  - An index
  - An anchor to an element
  - A set

CD
{from c: CD}

title: string
performer:
anchor(Ctx(PerformerByCD(self)))
songs: Idx (SongbyCD(self))
availability:integer
cover: image*
publicationDate: date

Notation: Access Structures



- Simple Access Structure

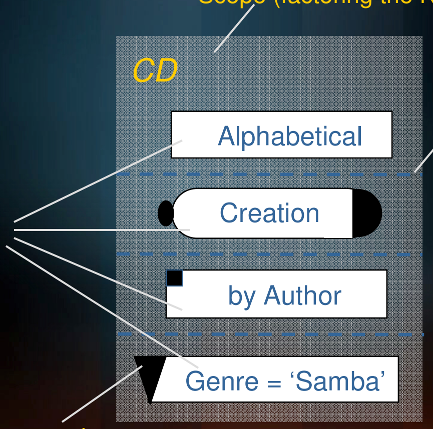
- Access structure with multiple ordering

- Dynamic Access structure – the query is defined at navigation time

- Hierarchical access structure – Selection in one level determines the elements of the next level

Notation: Navigation Context

Scope (factoring the Navigation Class)



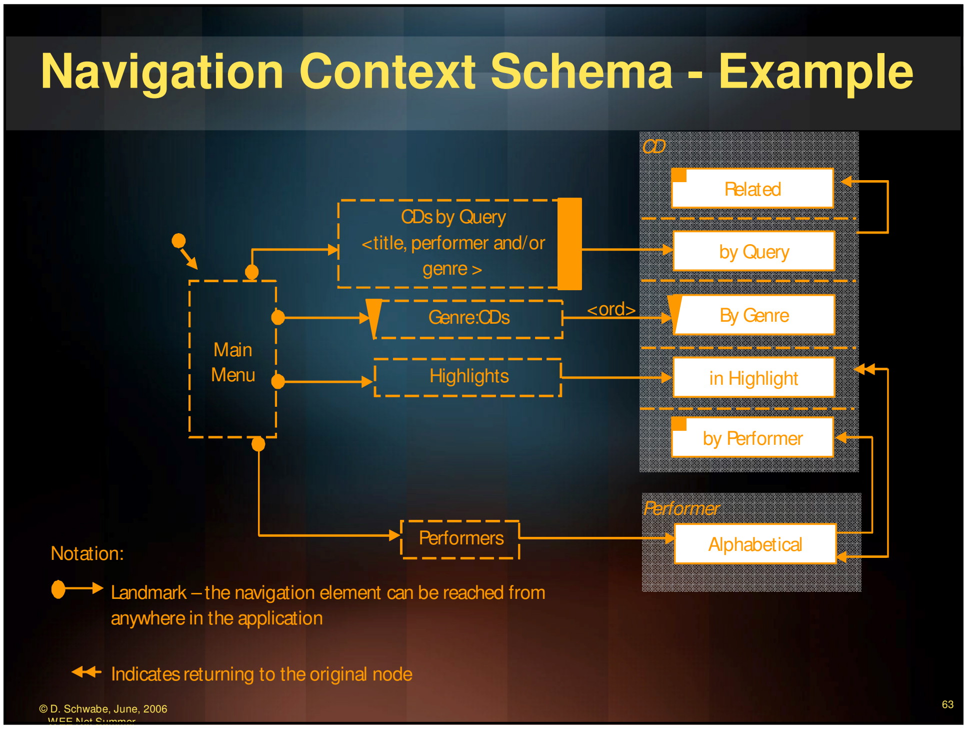
Indicates it is not possible to navigate among context outside the "partition"

Contexts

Context elements may be ordered in multiple ways

● - access to the context is protected

■ Context derived index



Navigation Context Schema - Example

CD

Related

by Query

By Genre

in Highlight

by Performer

CDs by Query <title, performer and/or genre >

Genre:CDs

<ord>

Highlights

Main Menu

Performers

Performer

Alphabetical

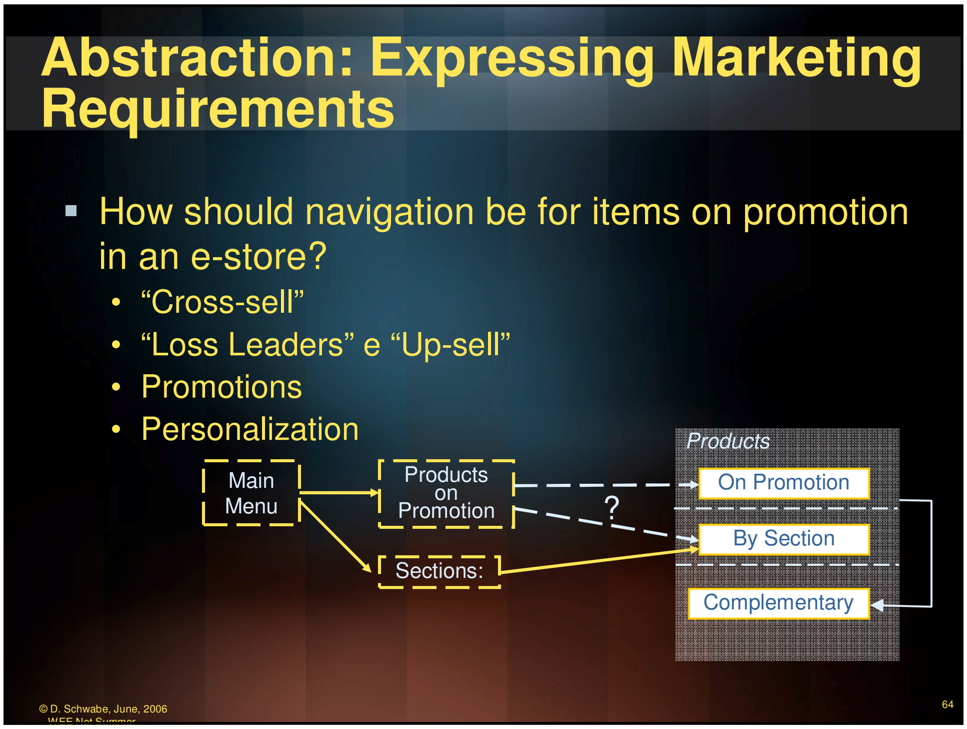
Notation:

●→ Landmark – the navigation element can be reached from anywhere in the application

◂◂ Indicates returning to the original node

© D. Schwabe, June, 2006

WEE Net Summer



Abstraction: Expressing Marketing Requirements

- How should navigation be for items on promotion in an e-store?
  - “Cross-sell”
  - “Loss Leaders” e “Up-sell”
  - Promotions
  - Personalization

Main Menu

Products on Promotion

Sections:

?

Products

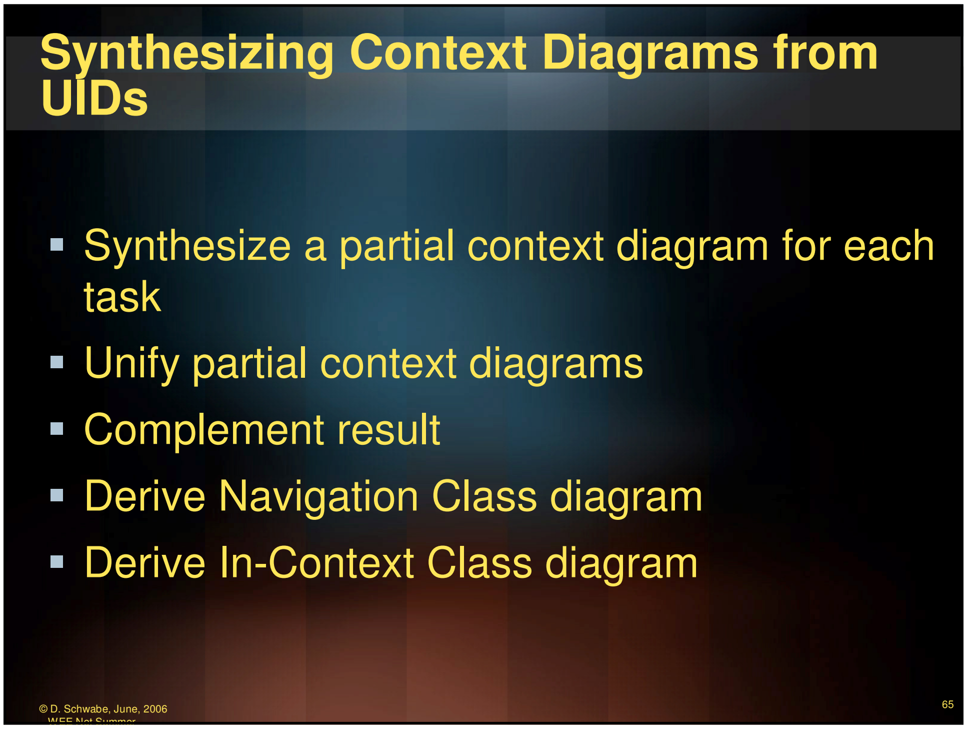
On Promotion

By Section

Complementary

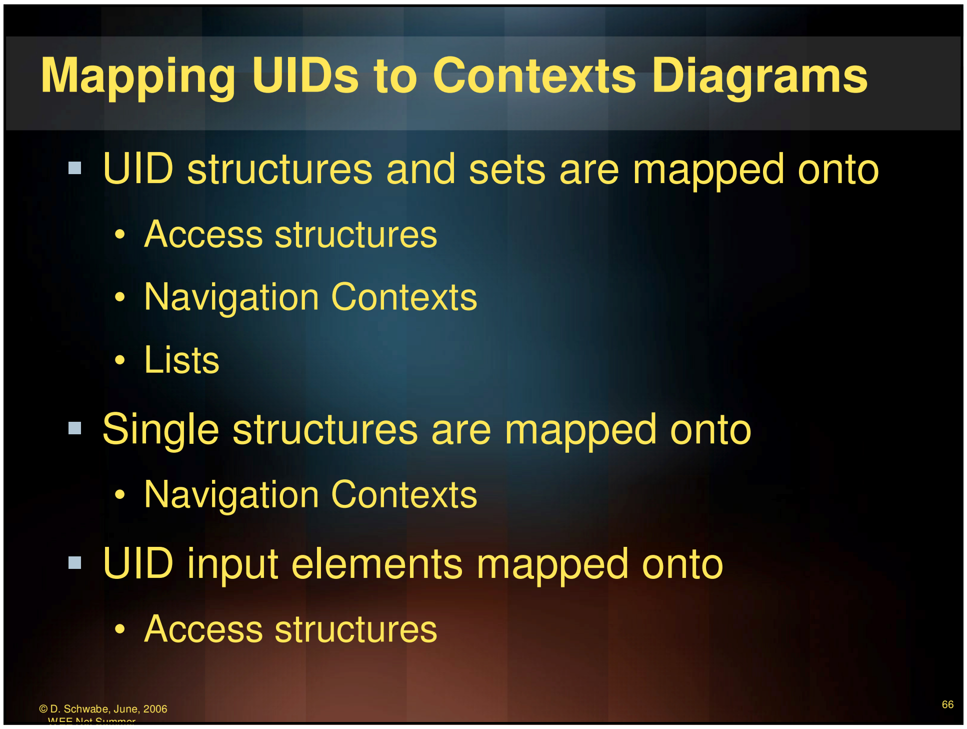
© D. Schwabe, June, 2006

WEE Net Summer



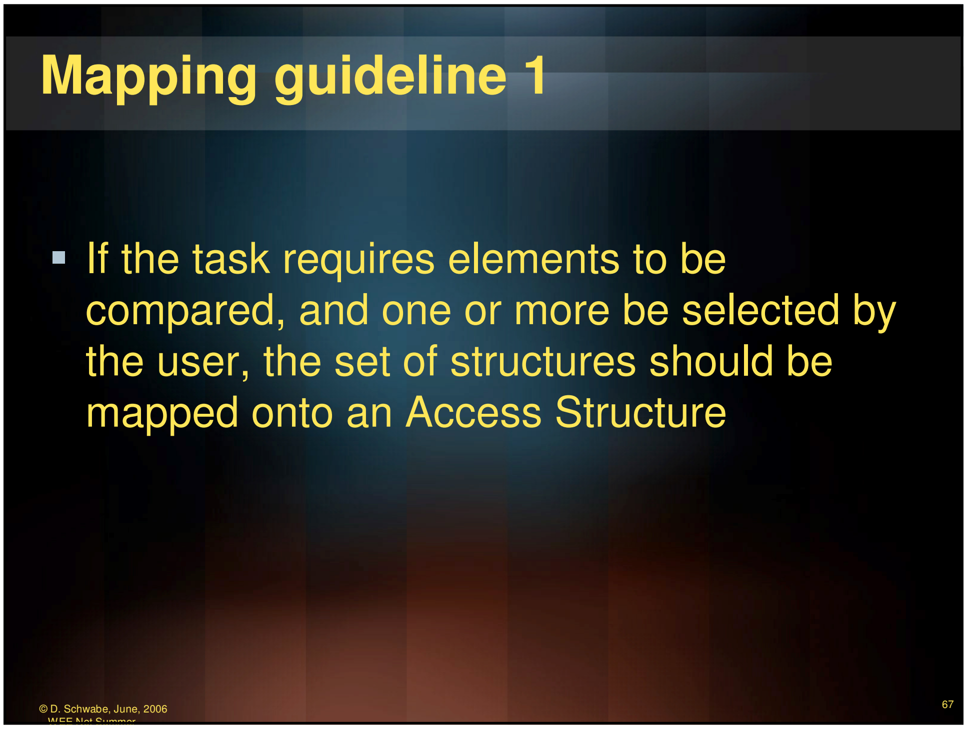
Synthesizing Context Diagrams from UIDs

- Synthesize a partial context diagram for each task
- Unify partial context diagrams
- Complement result
- Derive Navigation Class diagram
- Derive In-Context Class diagram



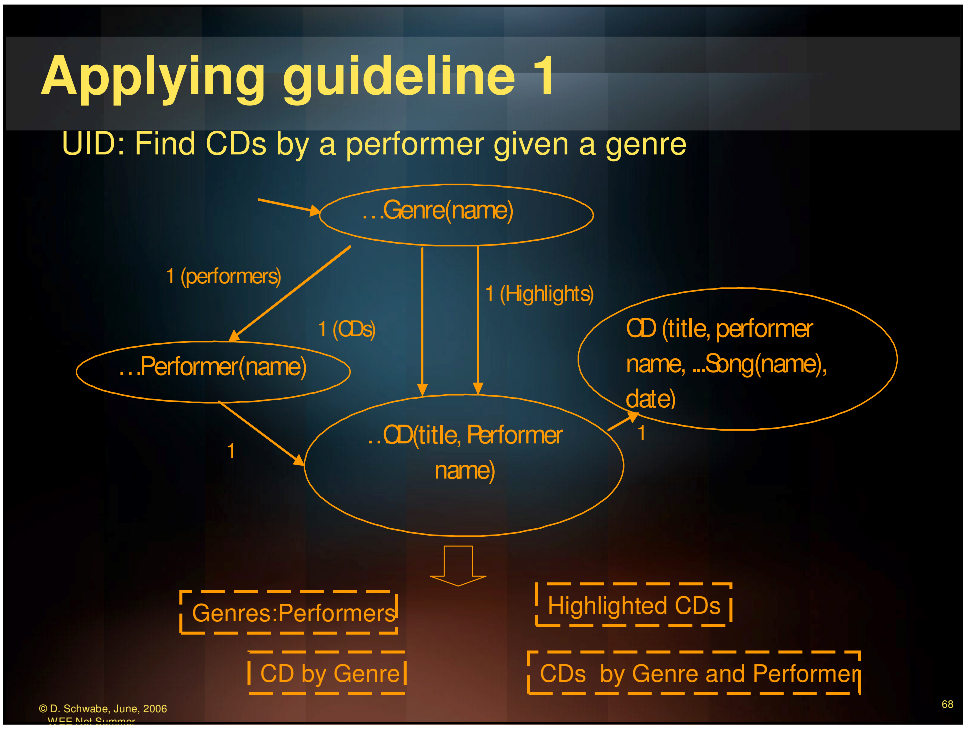
Mapping UIDs to Contexts Diagrams

- UID structures and sets are mapped onto
  - Access structures
  - Navigation Contexts
  - Lists
- Single structures are mapped onto
  - Navigation Contexts
- UID input elements mapped onto
  - Access structures



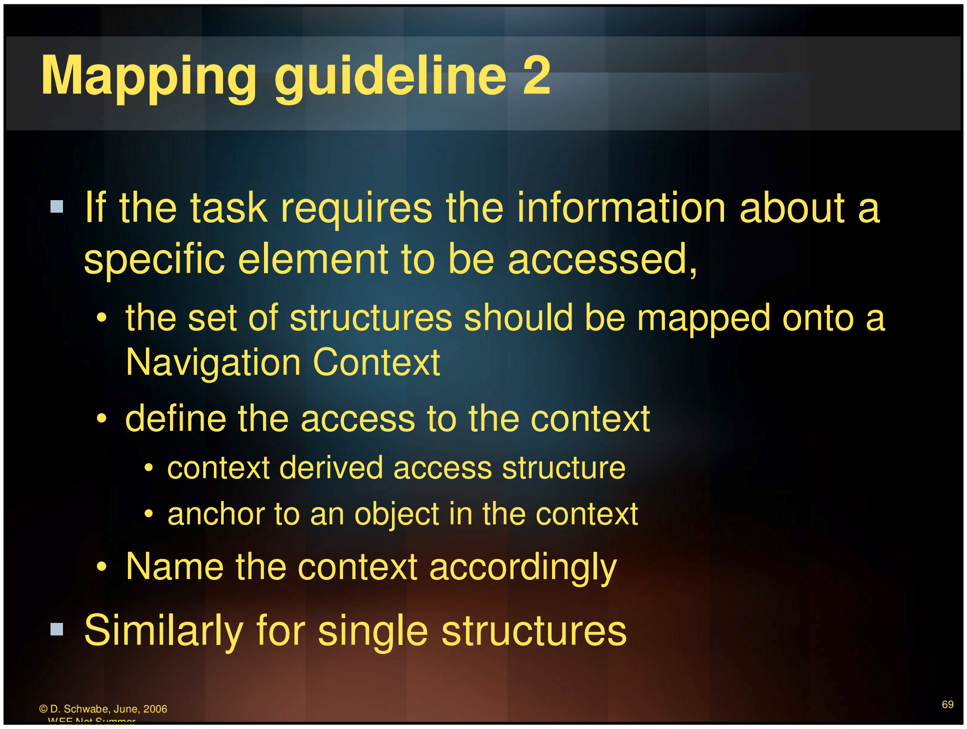
Mapping guideline 1

- If the task requires elements to be compared, and one or more be selected by the user, the set of structures should be mapped onto an Access Structure



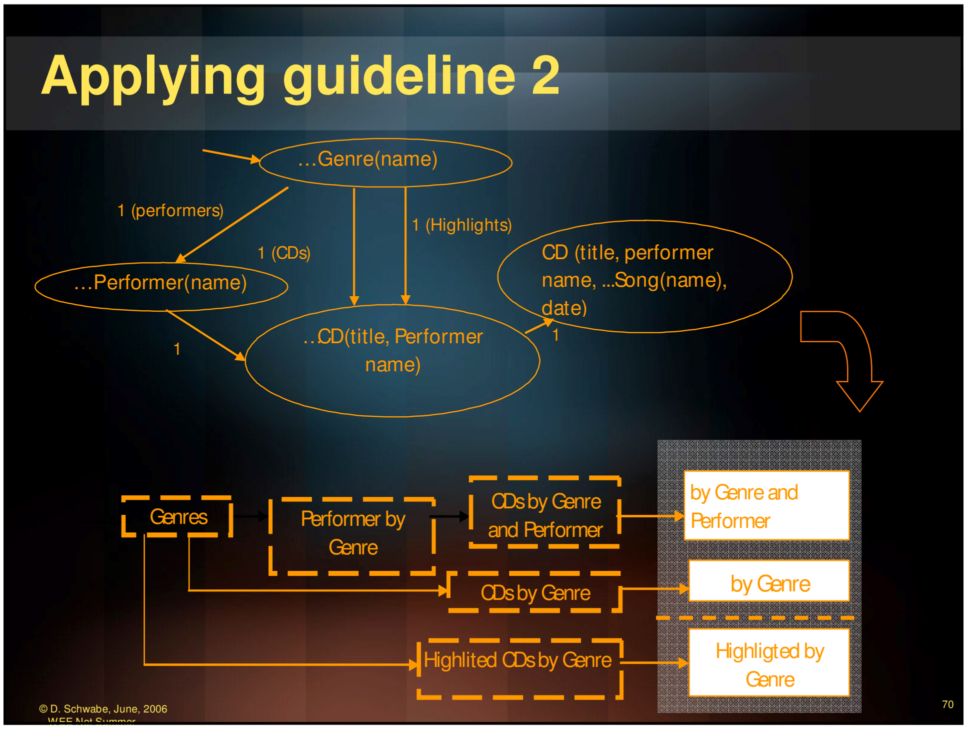
Applying guideline 1

UID: Find CDs by a performer given a genre



Mapping guideline 2

- If the task requires the information about a specific element to be accessed,
  - the set of structures should be mapped onto a Navigation Context
  - define the access to the context
    - context derived access structure
    - anchor to an object in the context
  - Name the context accordingly
- Similarly for single structures



Applying guideline 2

…Genre(name)

1 (performers)

1 (CDs)

1 (Highlights)

…Performer(name)

CD (title, performer name, …Song(name), date)

1

..CD(title, Performer name)

1

Genres

Performer by Genre

CDs by Genre and Performer

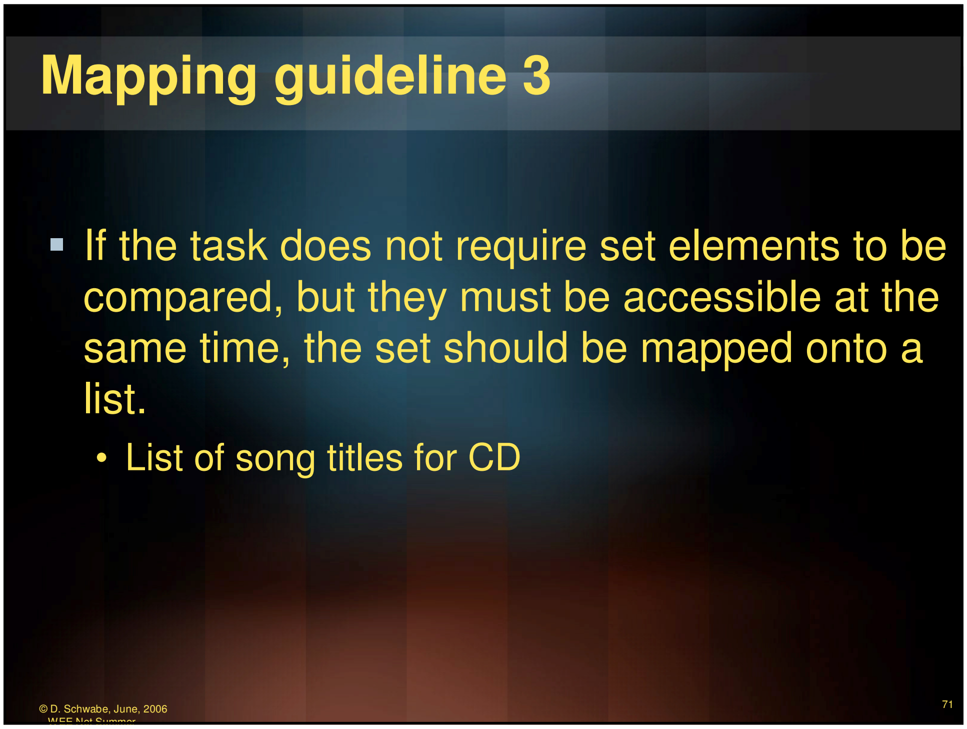
by Genre and Performer

CDs by Genre

by Genre

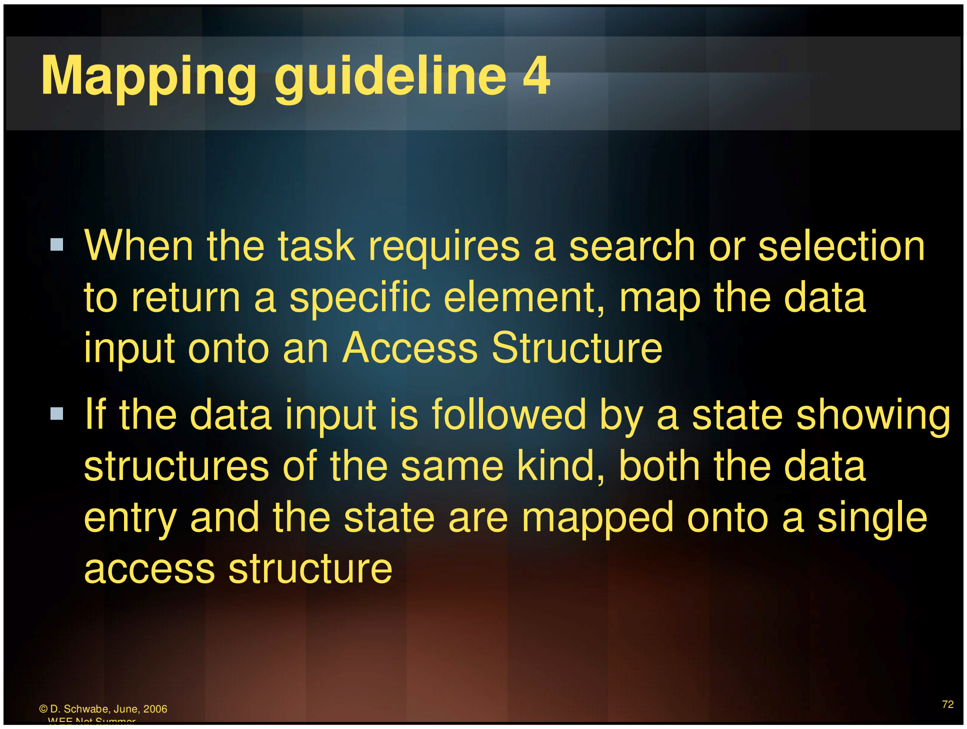
Highlited CDs by Genre

Highligted by Genre



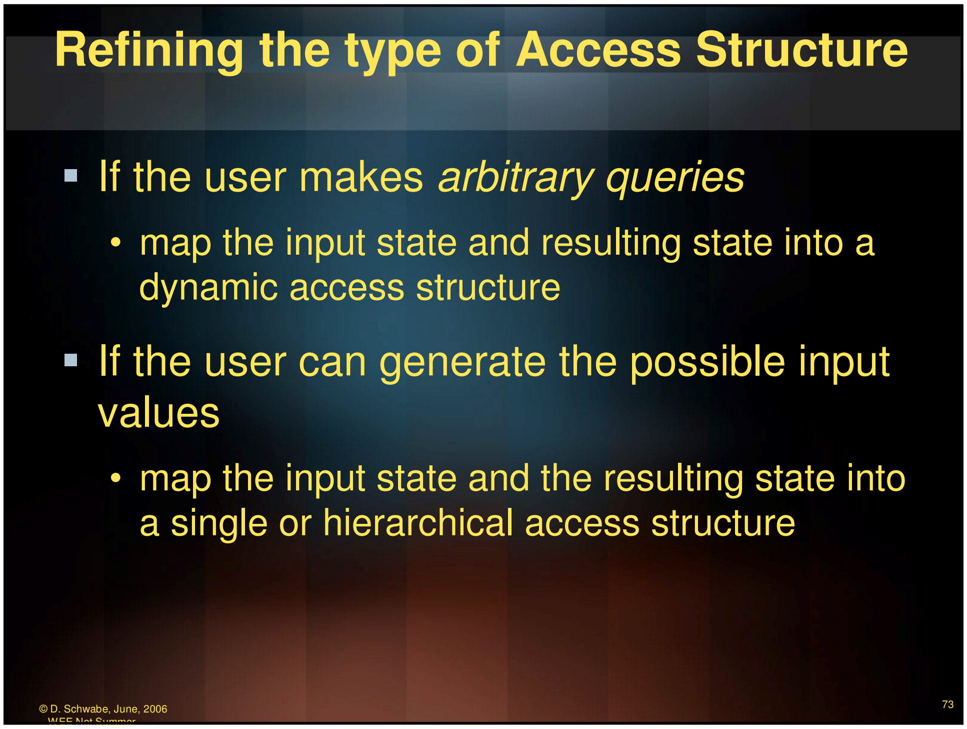
Mapping guideline 3

- If the task does not require set elements to be compared, but they must be accessible at the same time, the set should be mapped onto a list.
  - List of song titles for CD



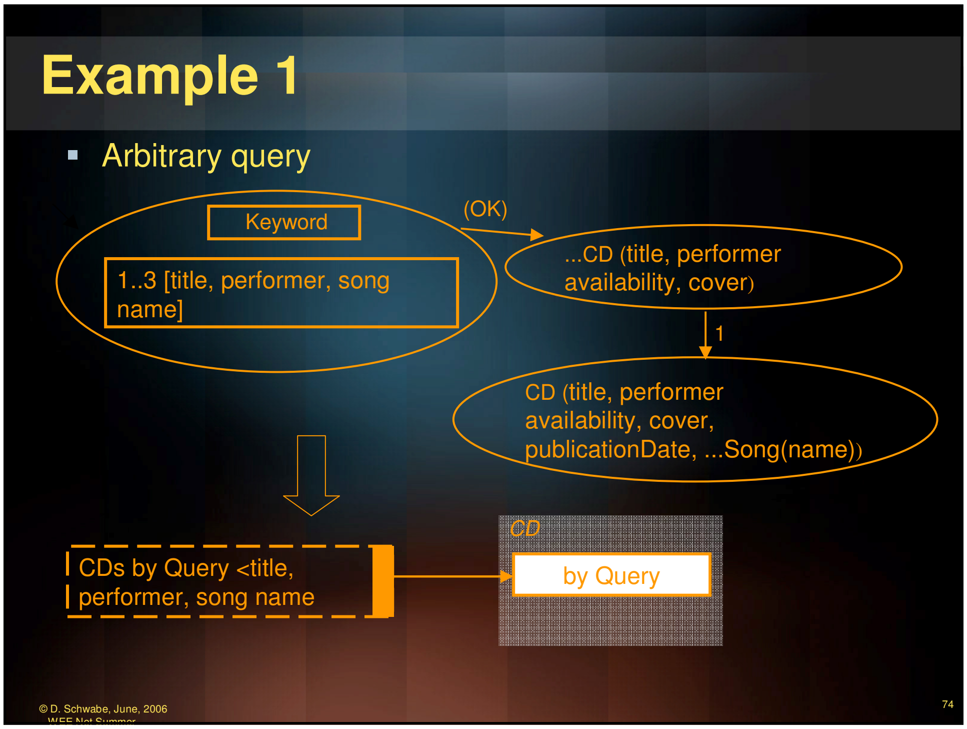
Mapping guideline 4

- When the task requires a search or selection to return a specific element, map the data input onto an Access Structure
- If the data input is followed by a state showing structures of the same kind, both the data entry and the state are mapped onto a single access structure



Refining the type of Access Structure

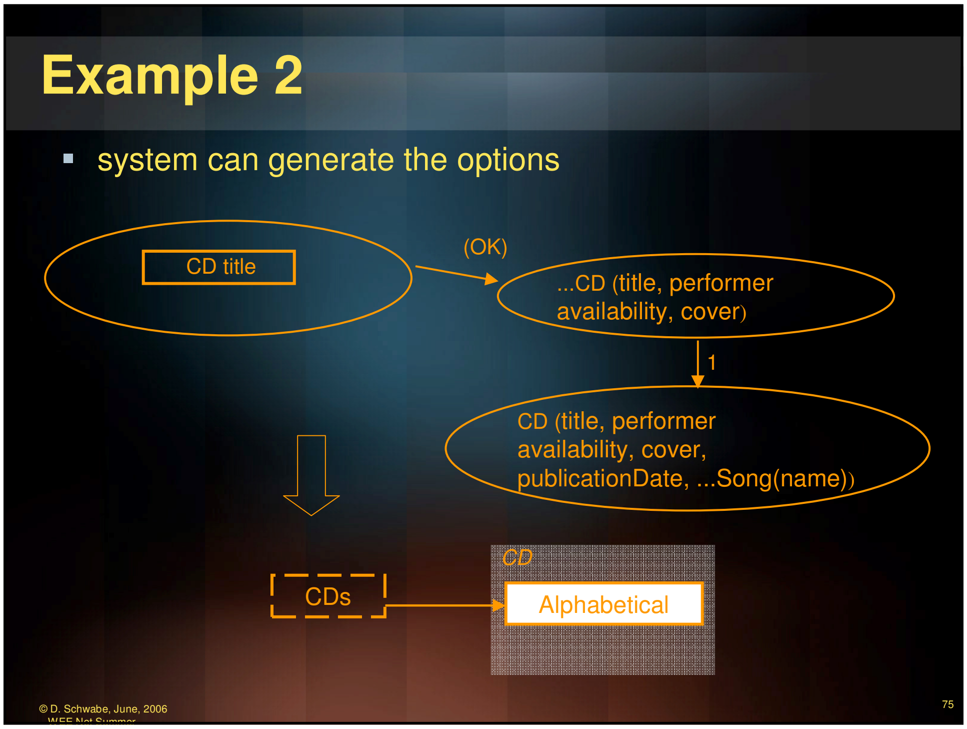
- If the user makes *arbitrary queries*
  - map the input state and resulting state into a dynamic access structure
- If the user can generate the possible input values
  - map the input state and the resulting state into a single or hierarchical access structure



Example 1

- Arbitrary query

Keyword
1..3 [title, performer, song name]
(OK)
...CD (title, performer availability, cover)
1
CD (title, performer availability, cover, publicationDate, ...Song(name))
CDs by Query <title, performer, song name
CD
by Query



Example 2

- system can generate the options

CD title

(OK)

...CD (title, performer availability, cover)

1

CD (title, performer availability, cover, publicationDate, ...Song(name))

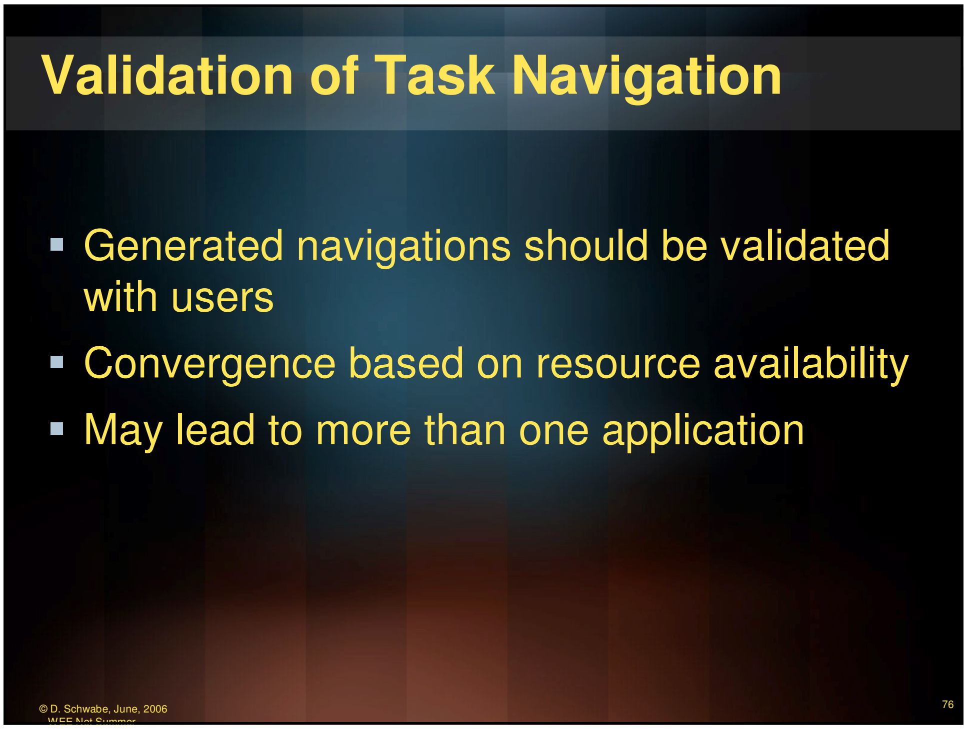
CDs

CD

Alphabetical

© D. Schwabe, June, 2006
WFE Net Summer

75

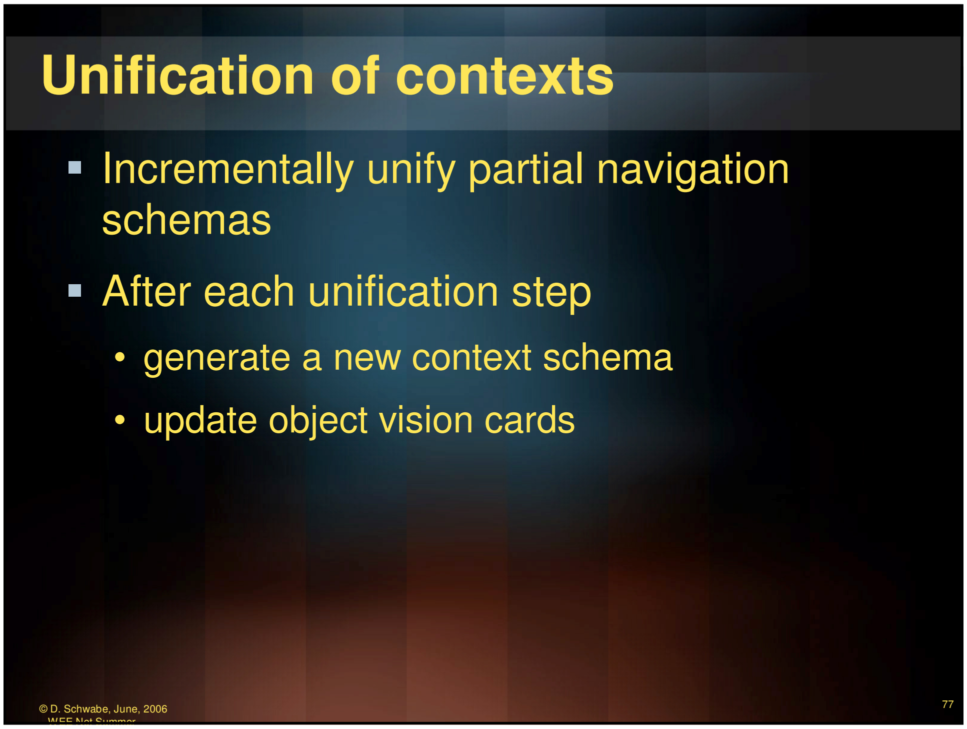


Validation of Task Navigation

- Generated navigations should be validated with users
- Convergence based on resource availability
- May lead to more than one application

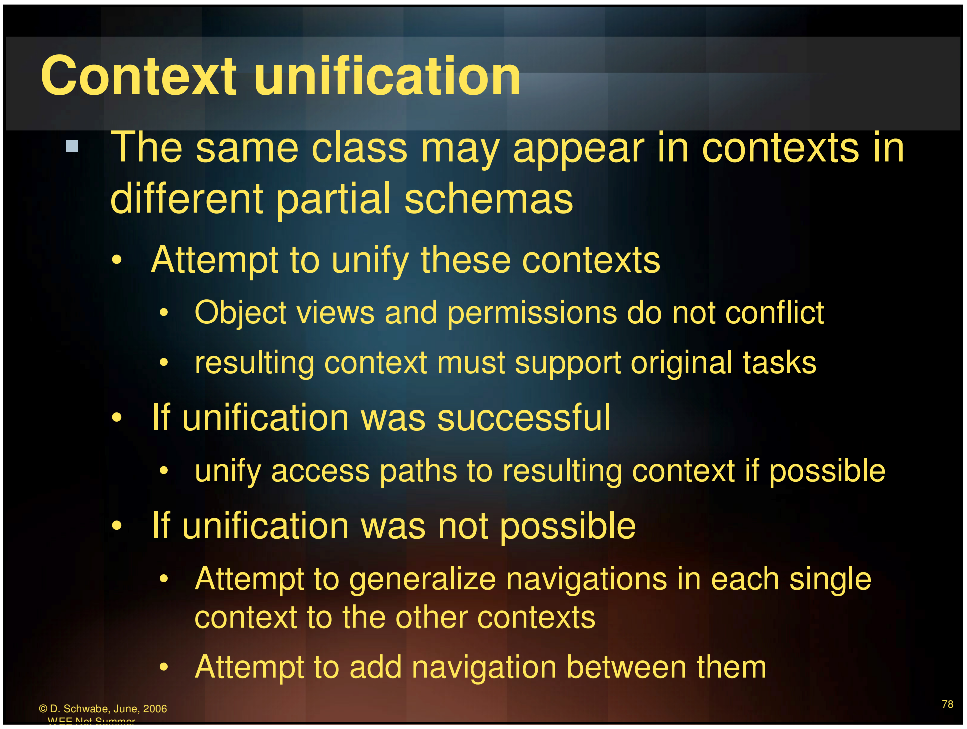
© D. Schwabe, June, 2006
WFE Net Summer

76



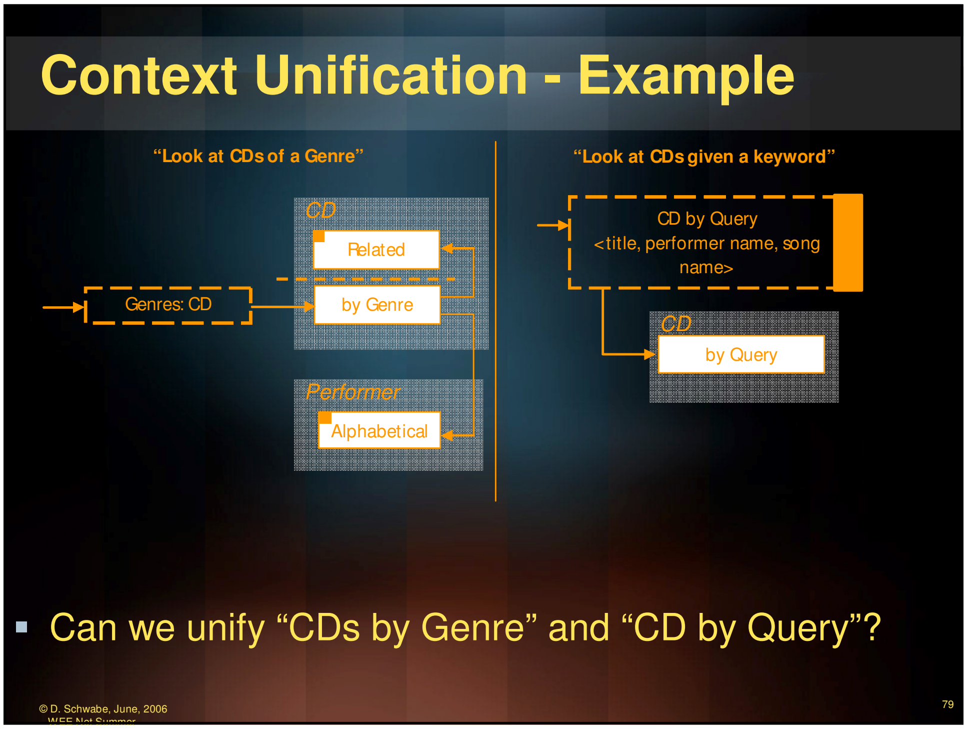
Unification of contexts

- Incrementally unify partial navigation schemas
- After each unification step
  - generate a new context schema
  - update object vision cards



Context unification

- The same class may appear in contexts in different partial schemas
  - Attempt to unify these contexts
    - Object views and permissions do not conflict
    - resulting context must support original tasks
  - If unification was successful
    - unify access paths to resulting context if possible
  - If unification was not possible
    - Attempt to generalize navigations in each single context to the other contexts
    - Attempt to add navigation between them



Context Unification - Example

"Look at CDs of a Genre"

CD

Related

Genres: CD

by Genre

Performer

Alphabetical

"Look at CDs given a keyword"

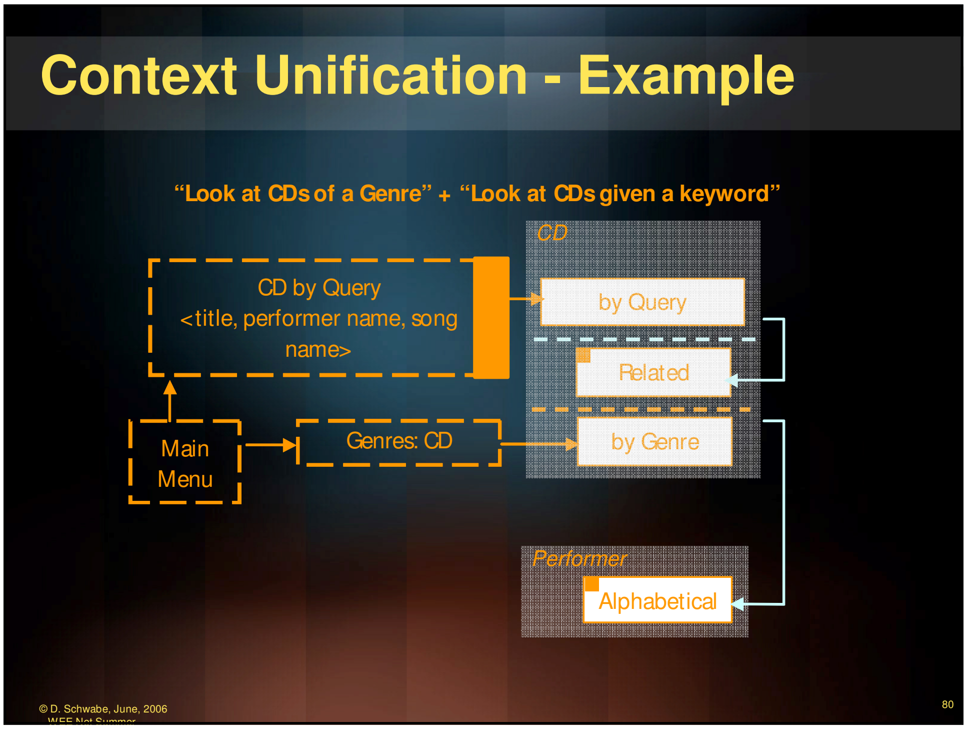
CD by Query
<title, performer name, song name>

CD

by Query

- Can we unify "CDs by Genre" and "CD by Query"?

© D. Schwabe, June, 2006
WEE Net Summer



Context Unification - Example

"Look at CDs of a Genre" + "Look at CDs given a keyword"

CD

CD by Query
<title, performer name, song name>

by Query

Related

Main Menu

Genres: CD

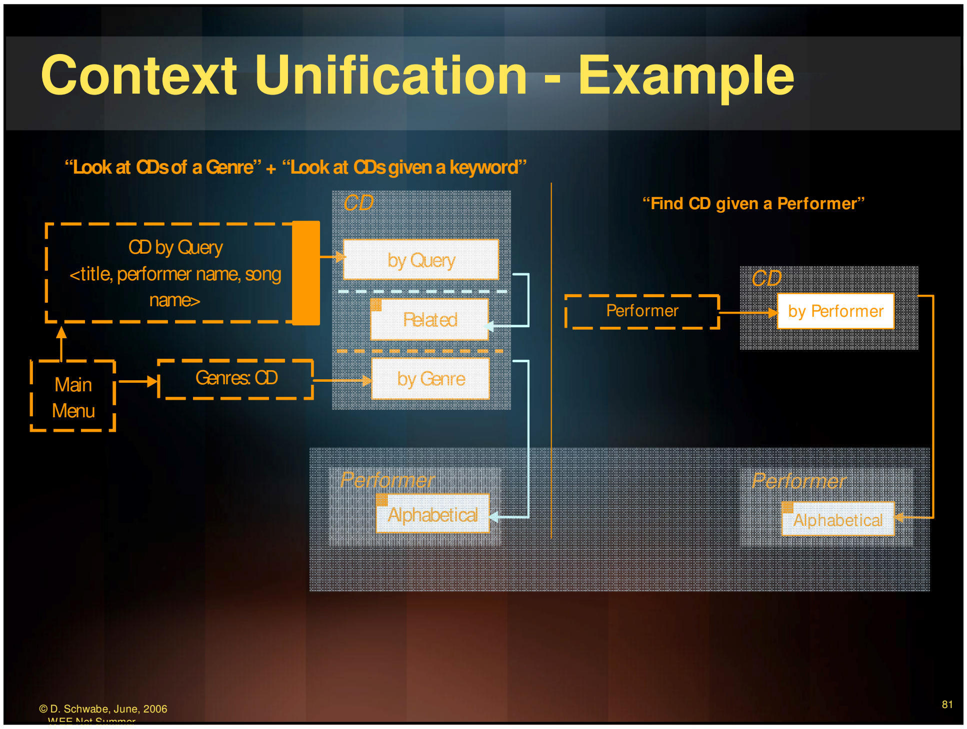
by Genre

Performer

Alphabetical

© D. Schwabe, June, 2006
WEE Net Summer

Context Unification - Example



"Look at CDs of a Genre" + "Look at CDs given a keyword"

CD

CD by Query
<title, performer name, song name>

by Query

Related

Main Menu

Genres: CD

by Genre

Performer

Alphabetical

"Find CD given a Performer"

Performer

CD

by Performer

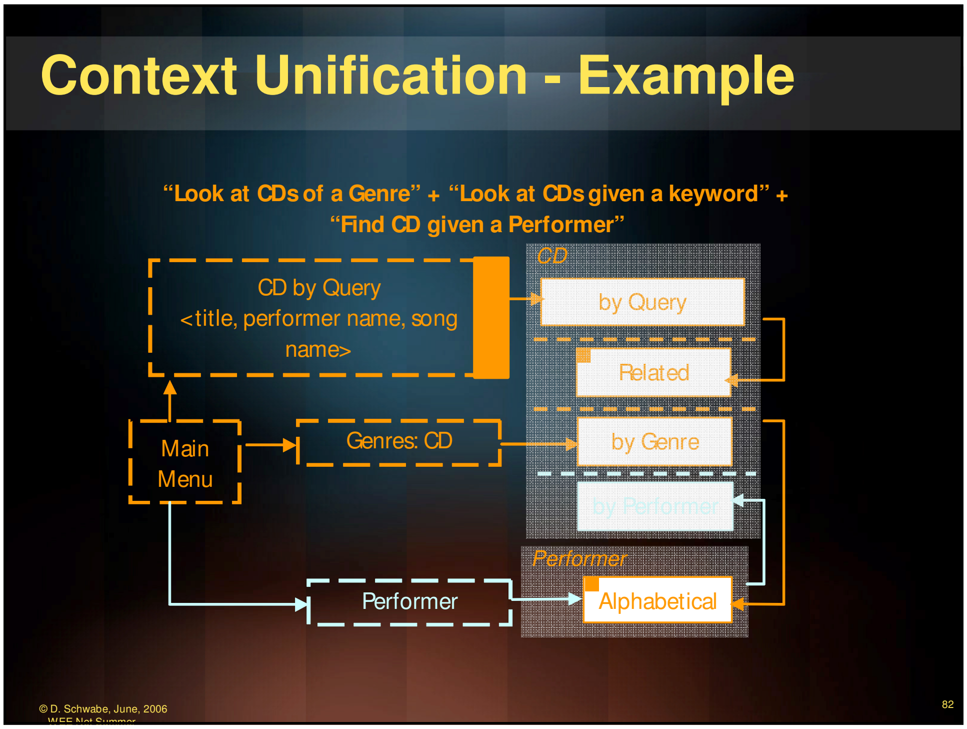
Performer

Alphabetical

© D. Schwabe, June, 2006
WEE Net Summer

81

Context Unification - Example



"Look at CDs of a Genre" + "Look at CDs given a keyword" +
"Find CD given a Performer"

CD

CD by Query
<title, performer name, song name>

by Query

Related

Main Menu

Genres: CD

by Genre

by Performer

Performer

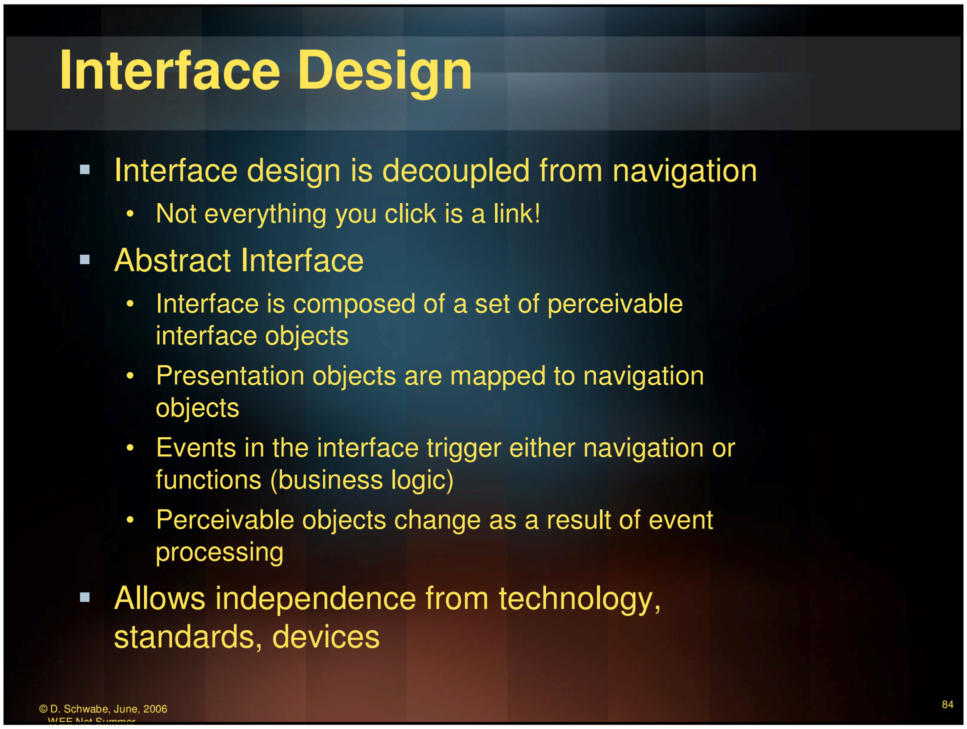
Performer

Alphabetical

© D. Schwabe, June, 2006
WEE Net Summer

82

Abstract Interface Design



Interface Design

- Interface design is decoupled from navigation
  - Not everything you click is a link!
- Abstract Interface
  - Interface is composed of a set of perceivable interface objects
  - Presentation objects are mapped to navigation objects
  - Events in the interface trigger either navigation or functions (business logic)
  - Perceivable objects change as a result of event processing
- Allows independence from technology, standards, devices

Interface Operations

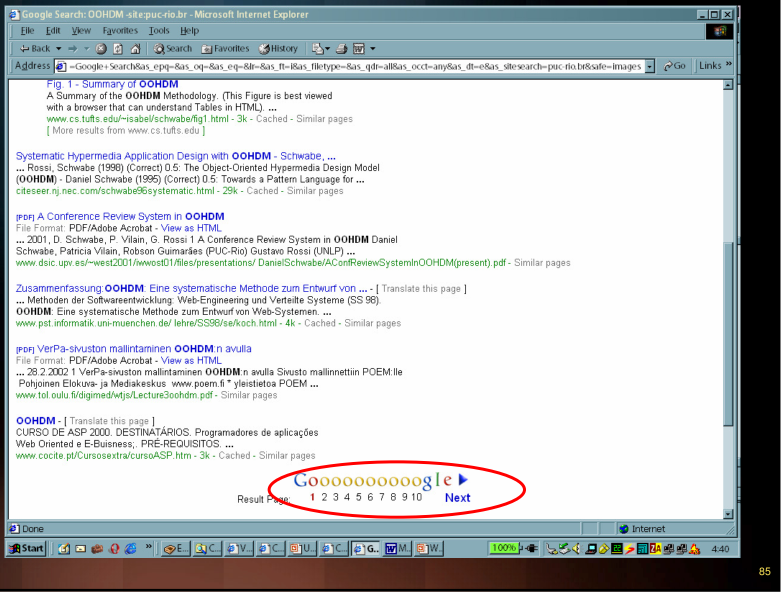

Google Search: OOHDM -site:puc-rio.br - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address =Google+Search&as_epq=&as_oq=&as_eq=&lr=&as_ft=i&as_filetype=&as_qdr=all&as_occt=any&as_dt=e&as_sitesearch=puc-rio.br&safe=images
Go
Links
Done
Internet
Start
E... C... V... C... U... C... G.. M.. W..
100%
4:40

Fig. 1 - Summary of **OOHDM**
A Summary of the **OOHDM** Methodology. (This Figure is best viewed with a browser that can understand Tables in HTML). ...
www.cs.tufts.edu/~isabel/schwabe/fig1.html - 3k - Cached - Similar pages
[More results from www.cs.tufts.edu]

Systematic Hypermedia Application Design with **OOHDM** - Schwabe, ...
... Rossi, Schwabe (1998) (Correct) 0.5: The Object-Oriented Hypermedia Design Model (**OOHDM**) - Daniel Schwabe (1995) (Correct) 0.5: Towards a Pattern Language for ...
citeseer.nj.nec.com/schwabe96systematic.html - 29k - Cached - Similar pages

[PDF] A Conference Review System in **OOHDM**
File Format: PDF/Adobe Acrobat - View as HTML
... 2001, D. Schwabe, P. Vilain, G. Rossi 1 A Conference Review System in **OOHDM** Daniel Schwabe, Patricia Vilain, Robson Guimarães (PUC-Rio) Gustavo Rossi (UNLP) ...
www.dsic.upv.es/~west2001/iwwost01/files/presentations/ DanielSchwabe/AConfReviewSystemInOOHDM(present).pdf - Similar pages

Zusammenfassung:**OOHDM**: Eine systematische Methode zum Entwurf von ... - [Translate this page]
... Methoden der Softwareentwicklung: Web-Engineering und Verteilte Systeme (SS 98). **OOHDM**: Eine systematische Methode zum Entwurf von Web-Systemen. ...
www.pst.informatik.uni-muenchen.de/ lehre/SS98/se/koch.html - 4k - Cached - Similar pages

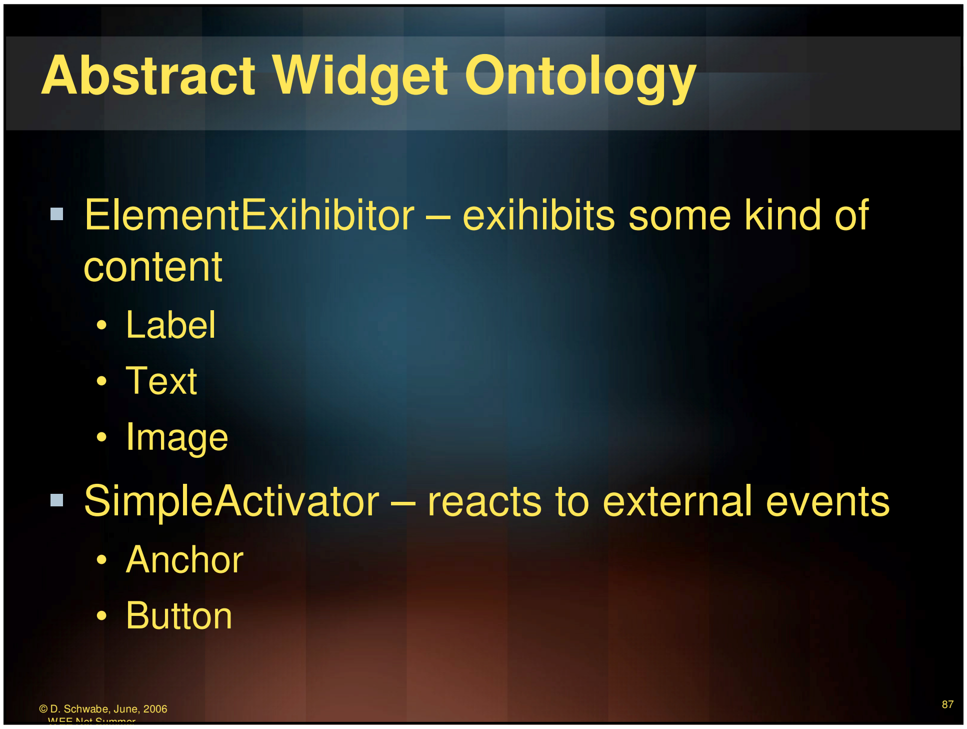
[PDF] VerPa-sivuston mallintaminen **OOHDM**:n avulla
File Format: PDF/Adobe Acrobat - View as HTML
... 28.2.2002 1 VerPa-sivuston mallintaminen **OOHDM**:n avulla Sivusto mallinnettiin POEM:lle Pohjoinen Elokuva- ja Mediakeskus www.poem.fi * yleistietoa POEM ...
www.tol.oulu.fi/digimed/wtjs/Lecture3oohdm.pdf - Similar pages

OOHDM - [Translate this page]
CURSO DE ASP 2000. DESTINATÁRIOS. Programadores de aplicações Web Oriented e E-Buisness;. PRÉ-REQUISITOS. ...
www.cocite.pt/Cursosextra/cursoASP.htm - 3k - Cached - Similar pages

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 Next

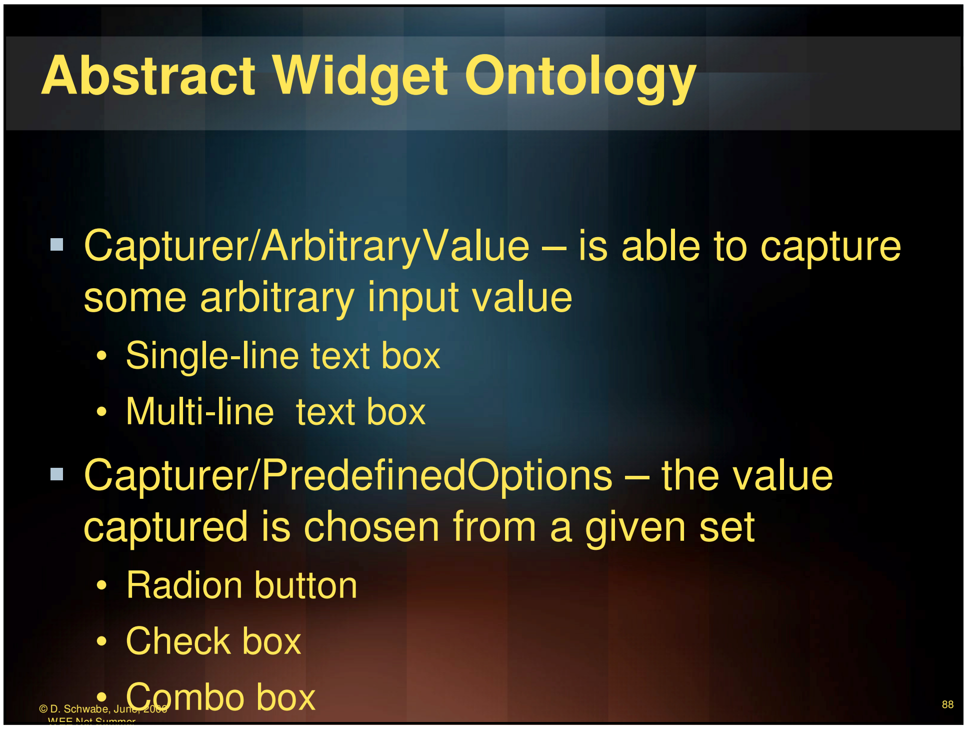
Proposed Approach

- Factor the interface specification in two level – abstract interface and concrete interface
- Abstract Widget Ontology
  - Describes interfaces focusing on information exchange aspects
- Concrete Widget Ontology
  - Describes concrete interface widgets commonly found in implementation environments
- Ontologies currently use OWL



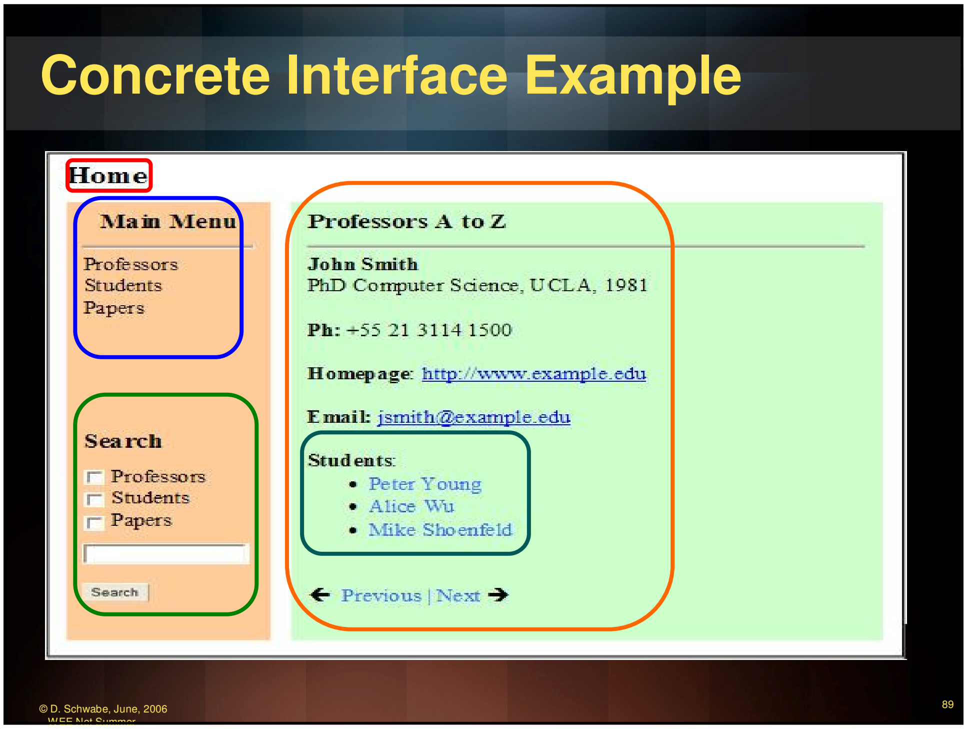
Abstract Widget Ontology

- ElementExihibitor – exihibits some kind of content
  - Label
  - Text
  - Image
- SimpleActivator – reacts to external events
  - Anchor
  - Button



Abstract Widget Ontology

- Capturer/ArbitraryValue – is able to capture some arbitrary input value
  - Single-line text box
  - Multi-line text box
- Capturer/PredefinedOptions – the value captured is chosen from a given set
  - Radion button
  - Check box
  - Combo box



Concrete Interface Example

Home

Main Menu

Professors
Students
Papers

Search

□ Professors
□ Students
□ Papers

Search

Professors A to Z

John Smith
PhD Computer Science, UCLA, 1981

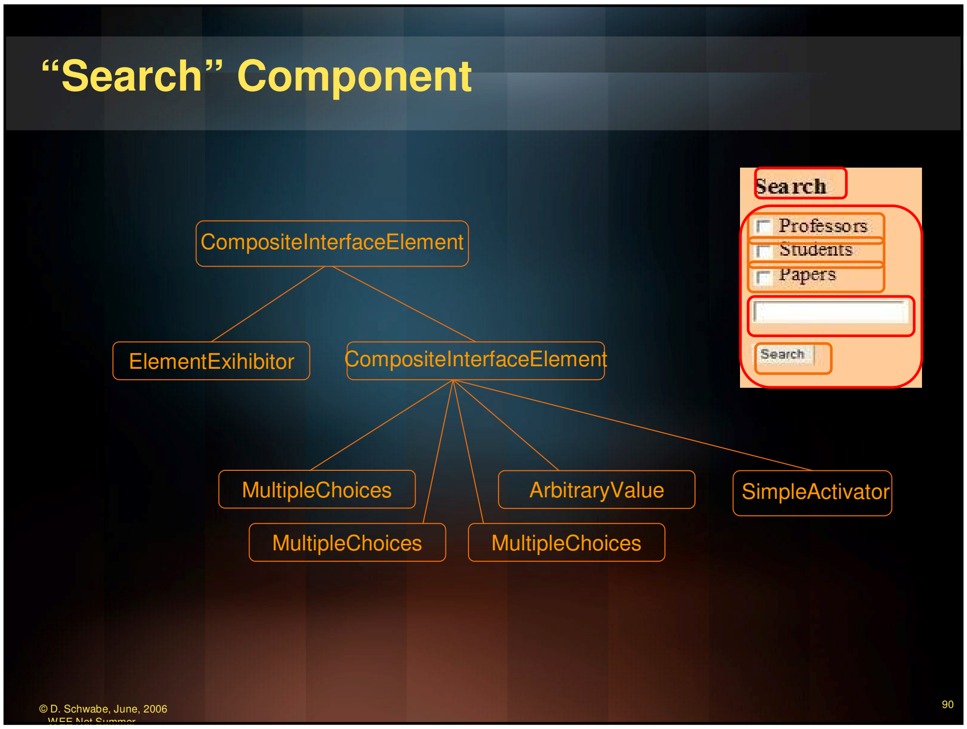
Ph: +55 21 3114 1500

Homepage: http://www.example.edu

Email: jsmith@example.edu

Students:
- Peter Young
- Alice Wu
- Mike Shoenfeld

Previous | Next



“Search” Component

CompositeInterfaceElement

ElementExihibitor

CompositeInterfaceElement

MultipleChoices

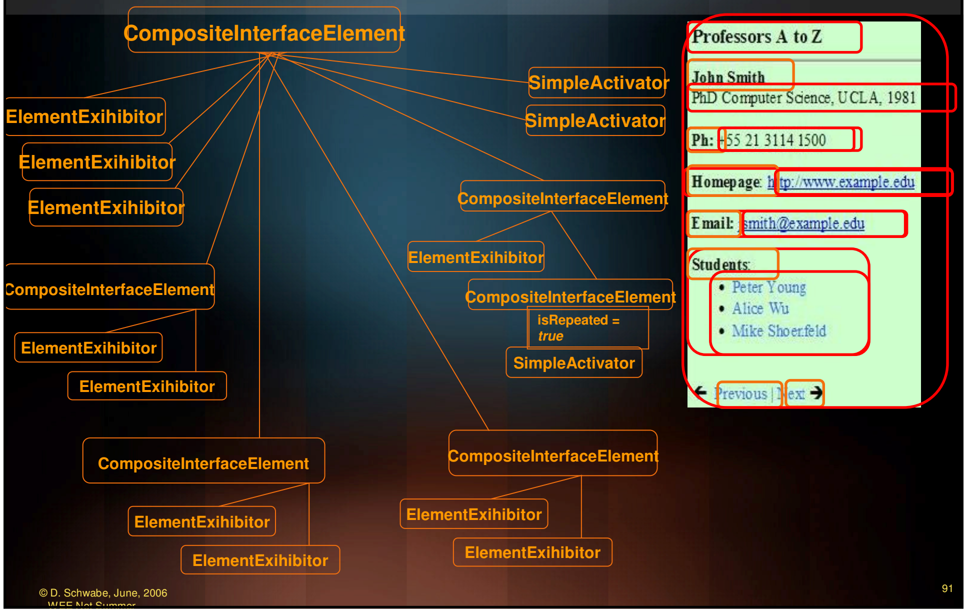
ArbitraryValue

SimpleActivator

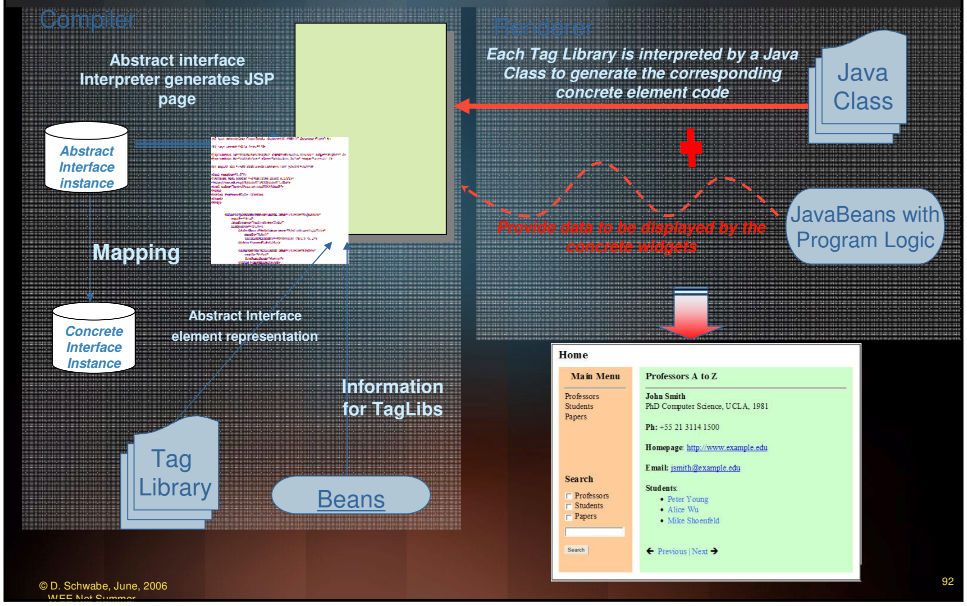
MultipleChoices

MultipleChoices

“Professor Data” Element



Interface Generation Environment


Compiler
Abstract interface Interpreter generates JSP page
Abstract Interface instance
Mapping
Concrete Interface Instance
Abstract Interface element representation
Information for TagLibs
Tag Library
Beans
Renderer
Each Tag Library is interpreted by a Java Class to generate the corresponding concrete element code
Java Class
Provide data to be displayed by the concrete widgets
JavaBeans with Program Logic

Home

Main Menu

Professors
Students
Papers

Search

□ Professors
□ Students
□ Papers

Search

Professors A to Z

John Smith
PhD Computer Science, UCLA, 1981

Ph: +55 21 3114 1500

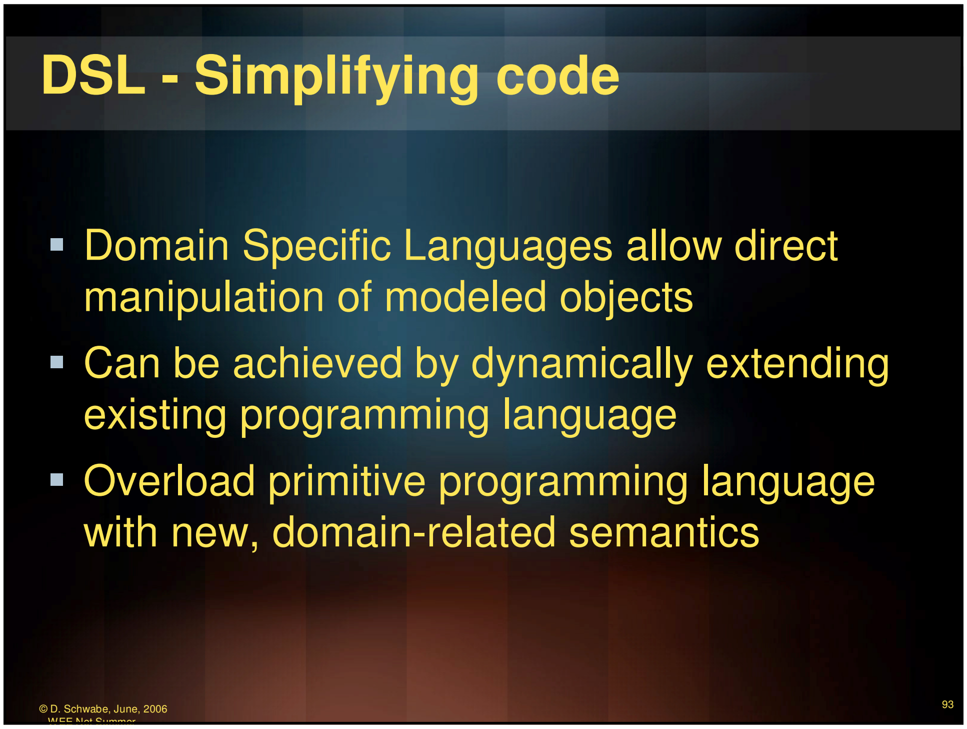
Homepage: http://www.example.edu

Email: jsmith@example.edu

Students:

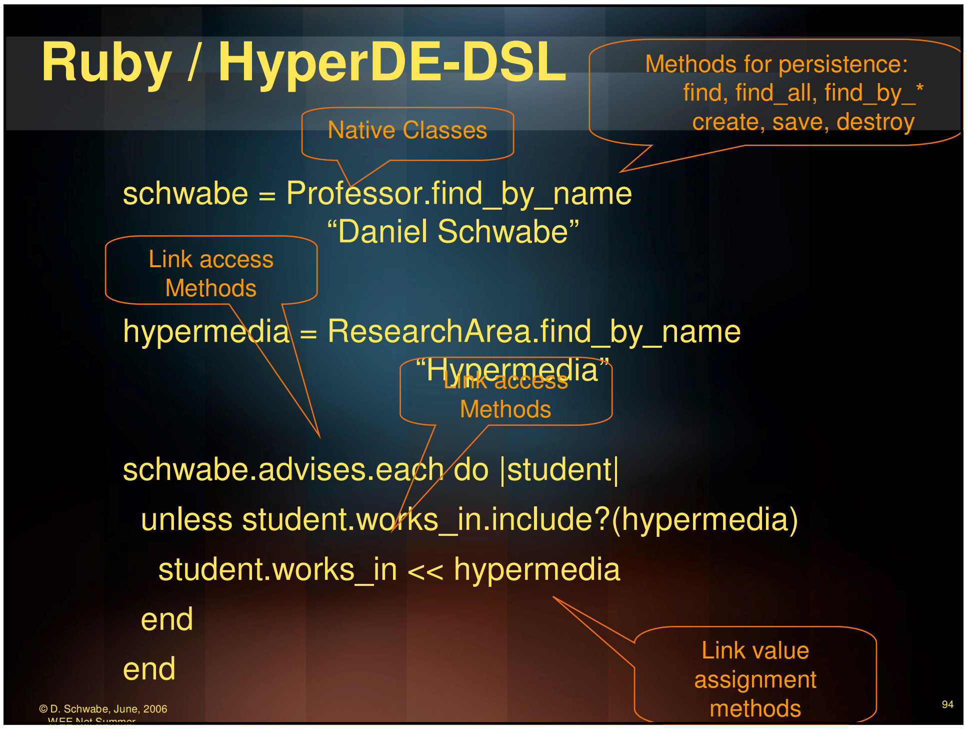
- Peter Young
- Alice Wu
- Mike Shoenfeld

← Previous | Next →



DSL - Simplifying code

- Domain Specific Languages allow direct manipulation of modeled objects
- Can be achieved by dynamically extending existing programming language
- Overload primitive programming language with new, domain-related semantics



Ruby / HyperDE-DSL

Methods for persistence:
find, find_all, find_by_*
create, save, destroy

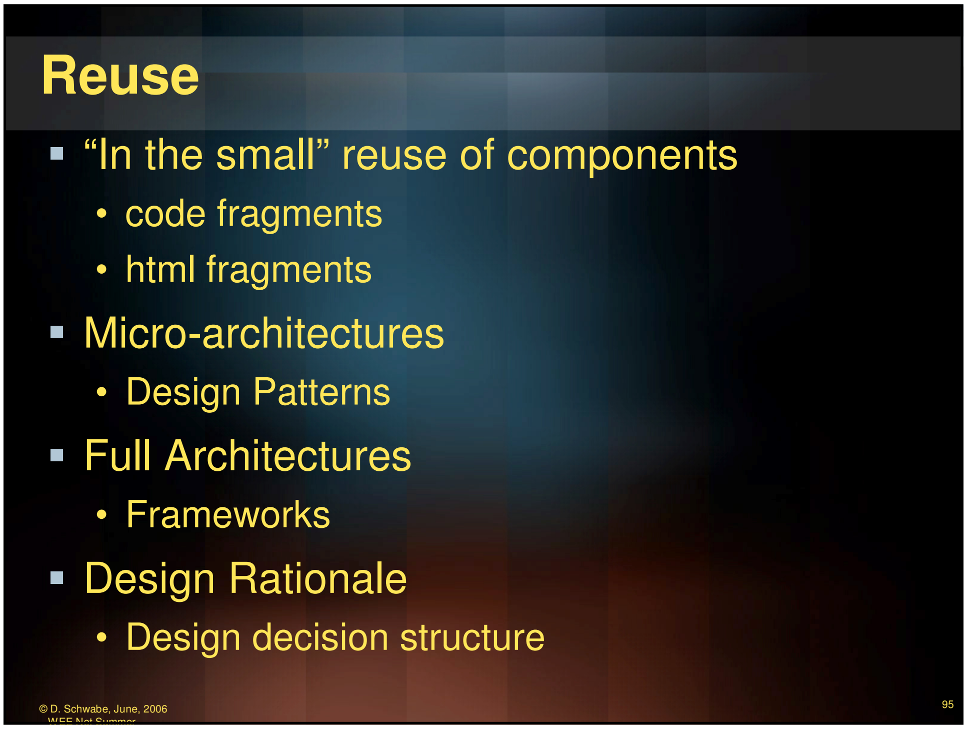
Native Classes

```
schwabe = Professor.find_by_name
             "Daniel Schwabe"

hypermedia = ResearchArea.find_by_name
                "Hypermedia"

schwabe.advises.each do |student|
  unless student.works_in.include?(hypermedia)
    student.works_in << hypermedia
  end
end
```

Link access
Methods

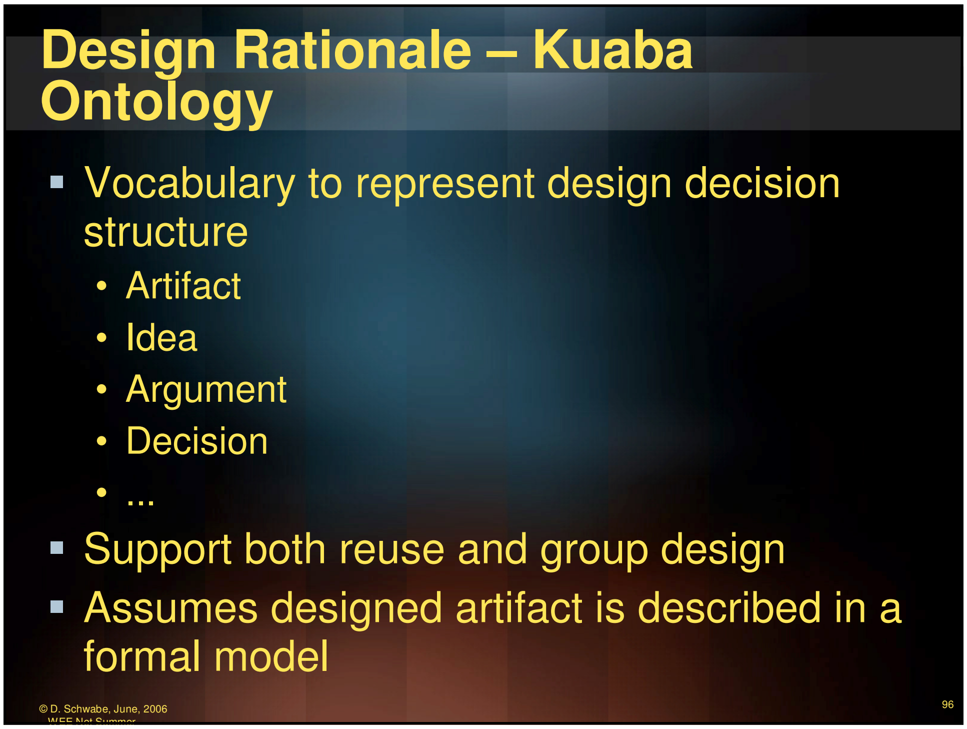
Link access
Methods

Link value
assignment
methods



Reuse

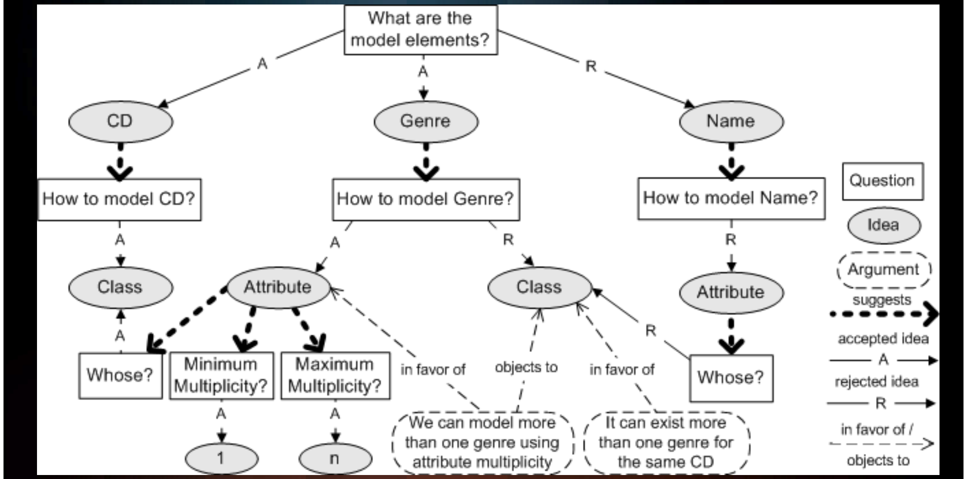
- "In the small" reuse of components
  - code fragments
  - html fragments
- Micro-architectures
  - Design Patterns
- Full Architectures
  - Frameworks
- Design Rationale
  - Design decision structure



Design Rationale – Kuaba Ontology

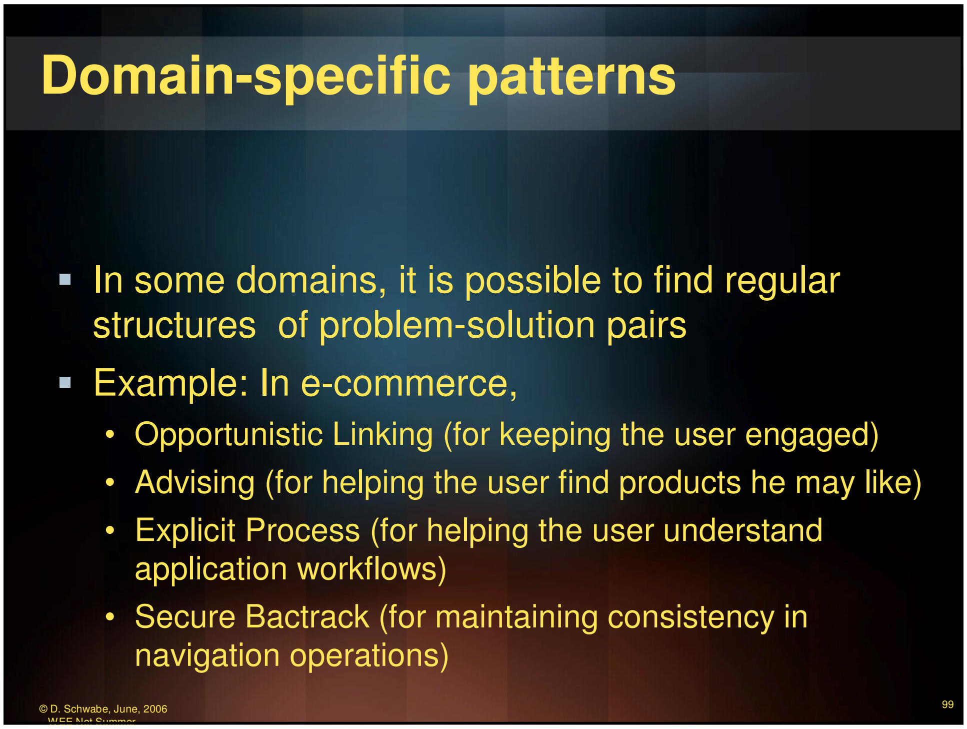
- Vocabulary to represent design decision structure
  - Artifact
  - Idea
  - Argument
  - Decision
  - ...
- Support both reuse and group design
- Assumes designed artifact is described in a formal model

Kuaba - Example



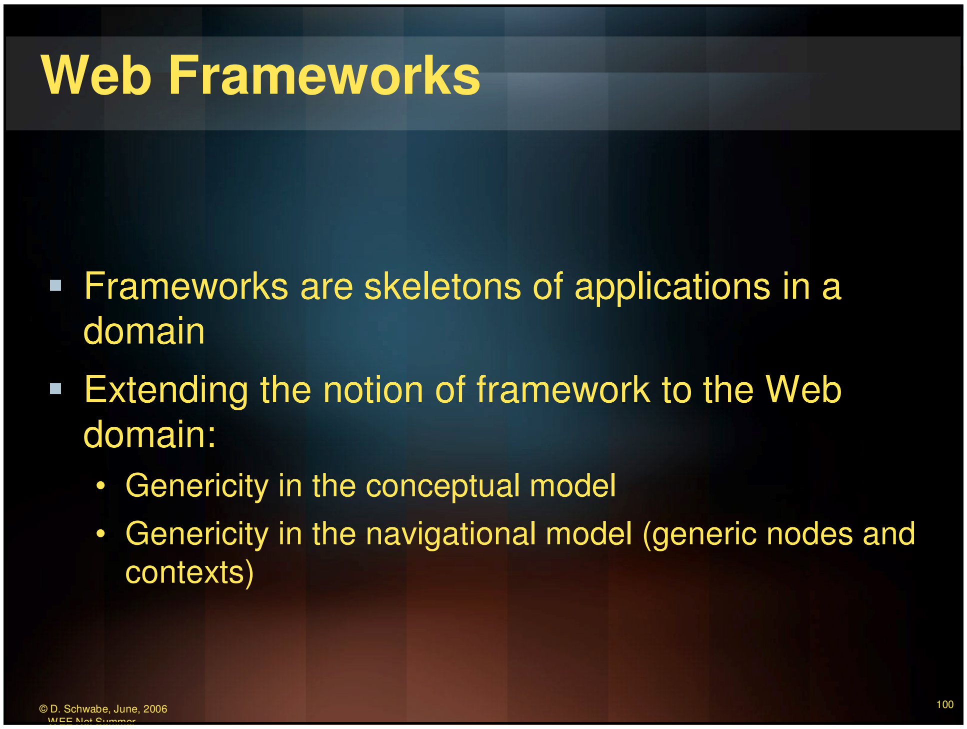
Navigation Patterns

- Help to record and convey good and recurrent navigation architectures
- Can be organized in catalogues and used as “books of experience”
- Examples:
  - Landmark (to access all important sub-sites)
  - News (to indicate new products)
  - Portal (to serve as a gateway to a set of services)
  - Set-Based Navigation



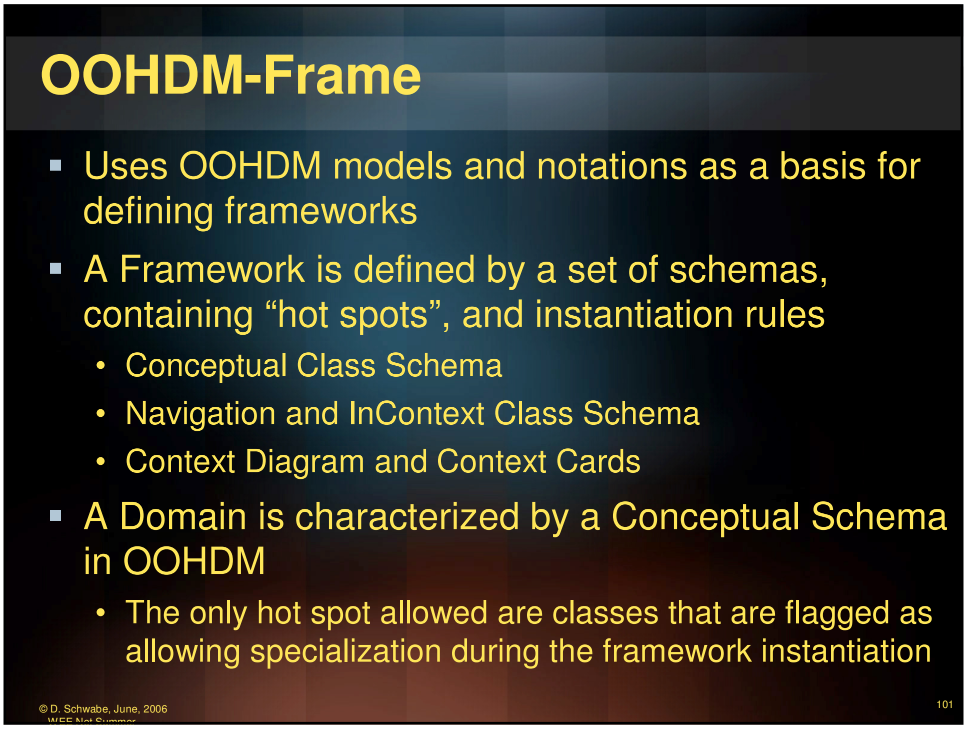
Domain-specific patterns

- In some domains, it is possible to find regular structures of problem-solution pairs
- Example: In e-commerce,
  - Opportunistic Linking (for keeping the user engaged)
  - Advising (for helping the user find products he may like)
  - Explicit Process (for helping the user understand application workflows)
  - Secure Bactrack (for maintaining consistency in navigation operations)



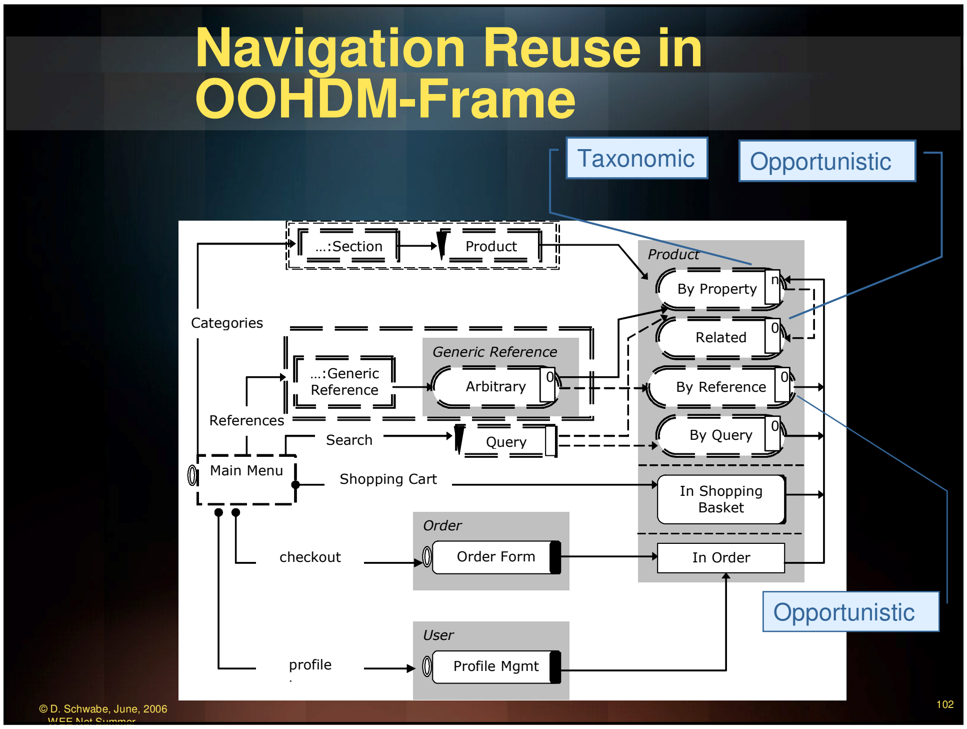
Web Frameworks

- Frameworks are skeletons of applications in a domain
- Extending the notion of framework to the Web domain:
  - Genericity in the conceptual model
  - Genericity in the navigational model (generic nodes and contexts)

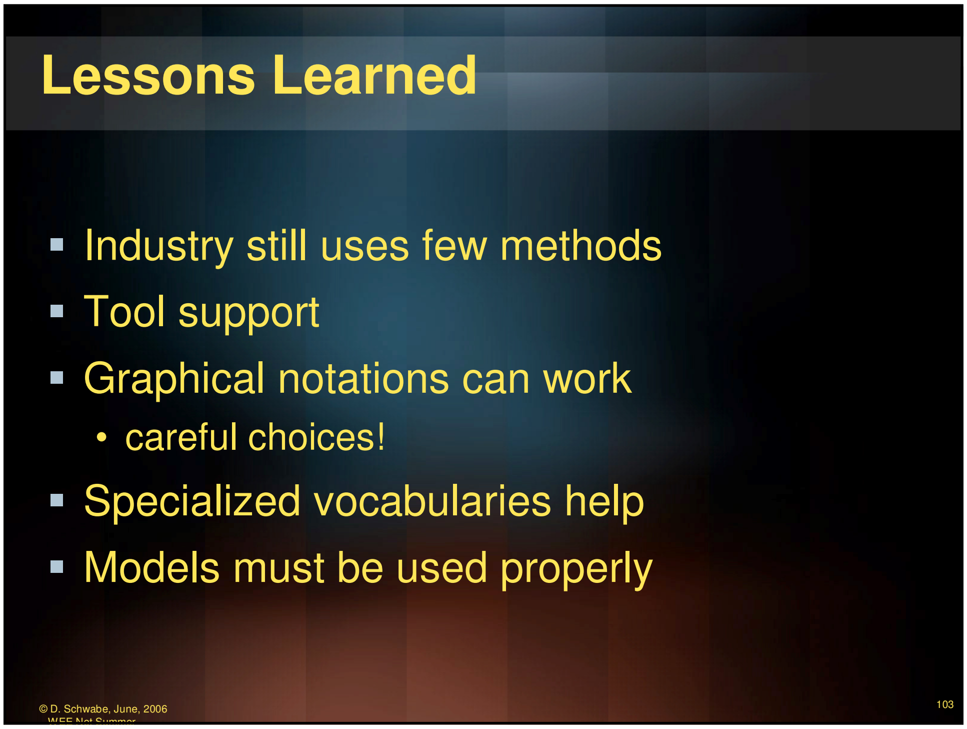


OOHDM-Frame

- Uses OOHDM models and notations as a basis for defining frameworks
- A Framework is defined by a set of schemas, containing “hot spots”, and instantiation rules
  - Conceptual Class Schema
  - Navigation and InContext Class Schema
  - Context Diagram and Context Cards
- A Domain is characterized by a Conceptual Schema in OOHDM
  - The only hot spot allowed are classes that are flagged as allowing specialization during the framework instantiation



Navigation Reuse in OOHDM-Frame

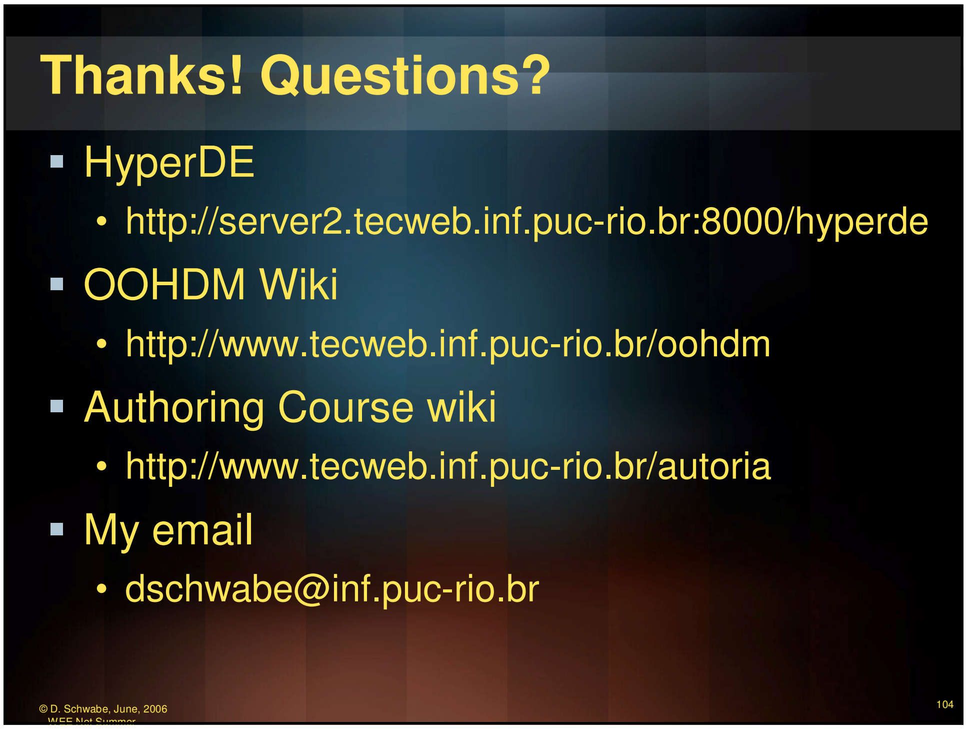


Lessons Learned

- Industry still uses few methods
- Tool support
- Graphical notations can work
  - careful choices!
- Specialized vocabularies help
- Models must be used properly

© D. Schwabe, June, 2006
WEE Net Summer

103



Thanks! Questions?

- HyperDE
  - http://server2.tecweb.inf.puc-rio.br:8000/hyperde
- OOHDM Wiki
  - http://www.tecweb.inf.puc-rio.br/oohdm
- Authoring Course wiki
  - http://www.tecweb.inf.puc-rio.br/autoria
- My email
  - dschwabe@inf.puc-rio.br

© D. Schwabe, June, 2006
WEE Net Summer

104