Web Engineering

Modelling Web Application (Part II)

Prof. Dr. Alexander Knapp

Dr. Nora Koch

SS 07 (4) – 04.06.07

Ludwig-Maximilians-Universität München

Contents

Overview of WebML

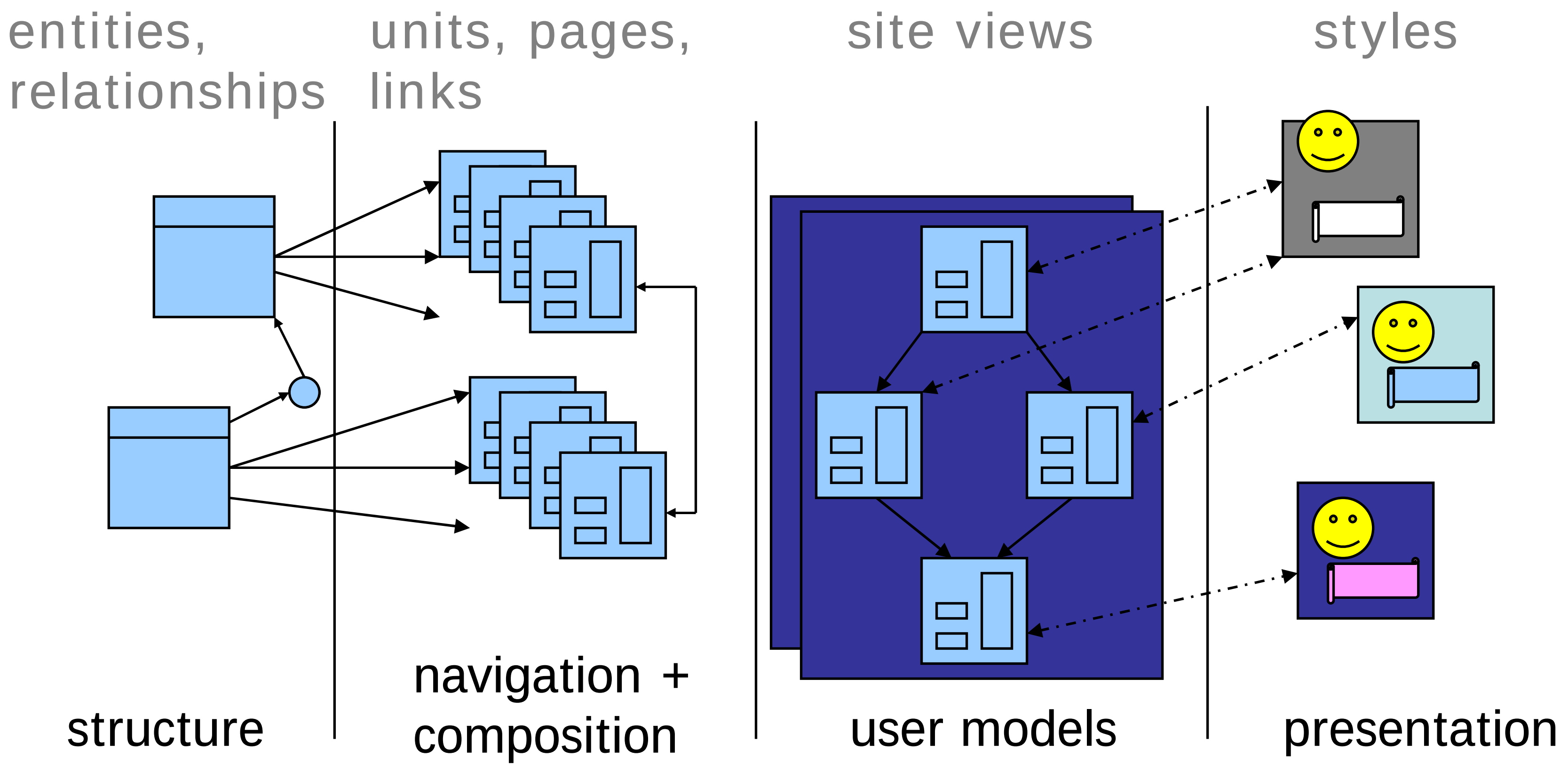
- **WebML**
  - provides graphical specifications for design of Web applications
  - embodied in a complete development process
  - which can be assisted by visual design tools
- **Target: data intensive Web sites**
  - large amount of data
  - interfaces directed to general public
    - exploratory
    - browsing-oriented
    - personalized
  - changing content, structure, navigation, presentation
- WebML is not the right approach for
  - small Web sites (homepages, …)
  - static Web sites

WebML Design Process

- **Main objectives**
  - expressing the structure of a Web application with a high-level description, which can be used for querying, evolution, and maintenance
  - providing multiple views of the same content
  - separating the information content from its composition into pages, navigation, and presentation, which can be defined and evolved independently
  - storing the meta-information collected during the design process within a repository, which can be used during the lifetime of the application for dynamically generating Web pages
  - modelling users and communities explicitly in the repository, to permit the specification of personalization policies and one-to-one applications
  - enabling the specification of data manipulation operations for updating the site content or interacting with arbitrary external services

Preview of WebML Concepts

- Site = Structure + Composition + Navigation + Presentation



WebML Models

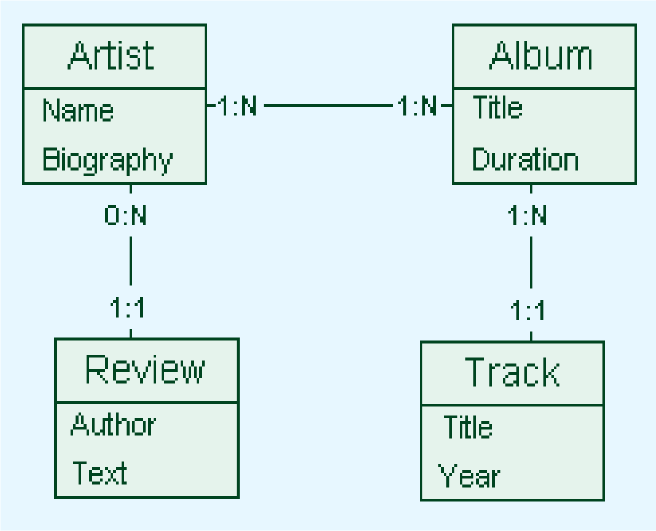
- **Structure/data**
  - data organization
- **Composition**
  - definition of site pages as set of subpages and elementary publishing units
- **Navigation**
  - links between pages
  - links between units
- **Presentation**
  - positioning of the units in the page
  - definition of graphical appearance

Data Model Elements

- **What are the objects published in the site and how are they related?**

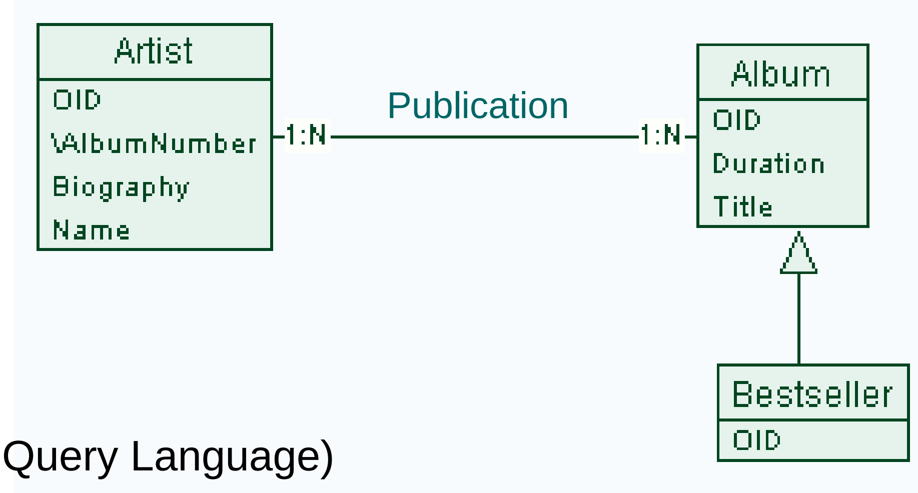
- **Fundamental elements**
  - entities: containers of data elements (object types)
  - attributes: scalar properties of an entity
  - relationships: semantic connections between entities
  - is-a hierarchy: classification and grouping

- Represented as a simplified
  - Entity-Relationship diagram
  - UML class diagrams



Modelling Data

- **Entities** can be organized in generalization hierarchies
- Instances of entities
  - individually addressable by means of a unique identifier (OID)
  - WebML OIDs are abstract concepts, which can be implemented in alternative ways in the underlying storage manager
    - primary keys in a relational data store
    - XML ID attributes in a XML data source



- **Relationships** can be restricted by means of cardinality constraints

- Entities, relationships and attributes can be **derived** (derivation model)
  - applied for redundant data
  - specified using a WebML-OQL (Object Query Language)
    - `Artist.AlbumNumber = count(self.Publication)`
    - `Bestseller := Album where Album.Sales > 1000000`

Hypertext Model

- **Hypertext modelling** specifies composition and navigation of the site
  - nodels at a high level the front-end of a dynamic Web application and the interactions with the back end business logic and data

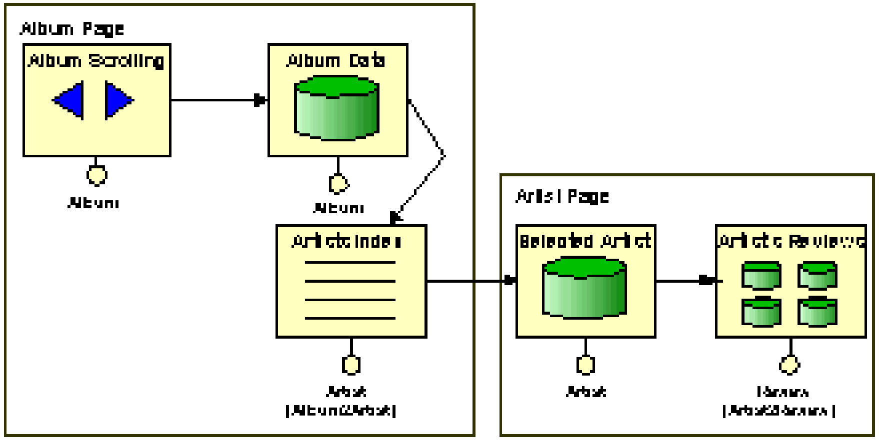
- **Composition**
  - describes which pages compose the hypertext and
  - which content units make up a page
  - pages of the Web site are the containers of information actually delivered to the reader
- **Navigation** of the site is specified through links
  - links can be defined between
    - the units inside a single page
    - units placed in different pages
    - pages

Hypertext Model Elements

- **What are the objects shown in the site and how can they be browsed?**

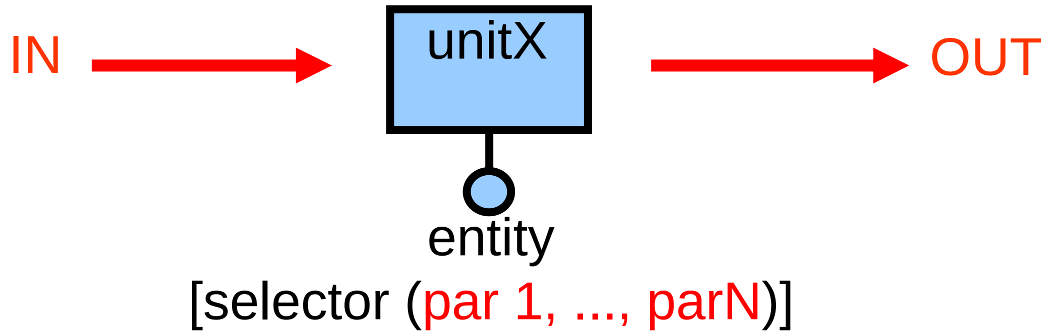
- **Fundamental elements**
  - content units: information published in the hypertext nodes
  - links: the way hypertext nodes are connected
  - siteviews: a set of pages that the user can experience as a whole Web site
  - areas: groups of pages devoted to a cohesive communication of business purpose within a site view
  - pages: units which are served to the user

- Represented by a proprietary notation
  - visual icons
  - semantics



Content Units (1)

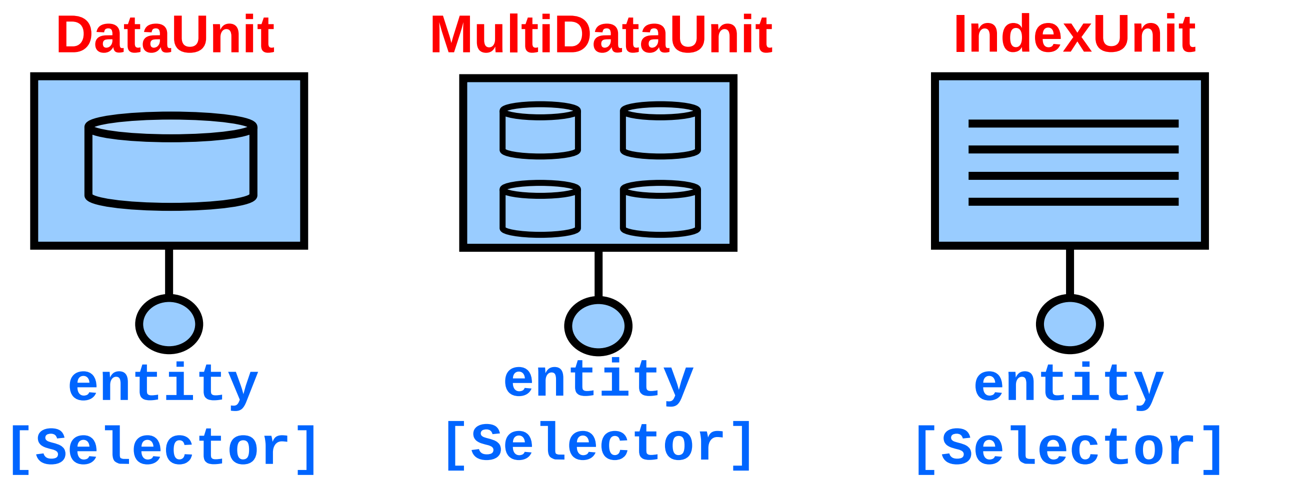
- A **WebML unit** is the atomic information publishing element

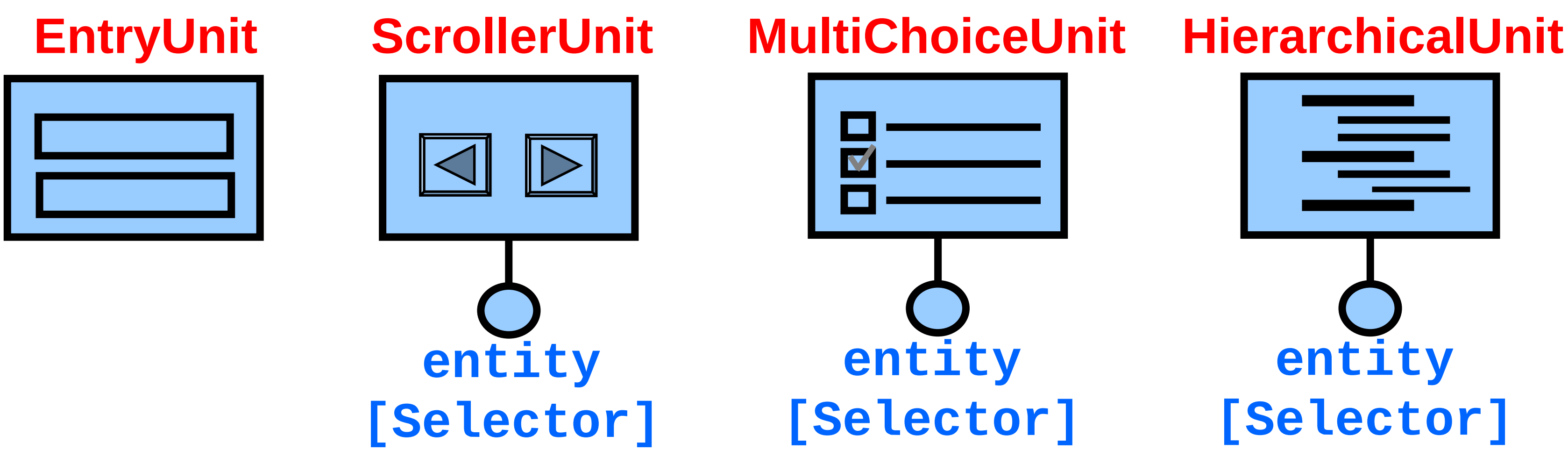


- It is a “**view**” defined upon a container of objects
  - all the instances of an entity
  - instances of an entity that meet a selection condition called selector
- Unit input and output
  - a unit may need some “context” to be computed
  - each unit exposes input and output parameters
  - input is required to compute the unit itself
    - parameters pre-defined for the unit +
    - other parameters required by the selector of the unit
  - output can be used to compute other unit(s) depending on the current unit

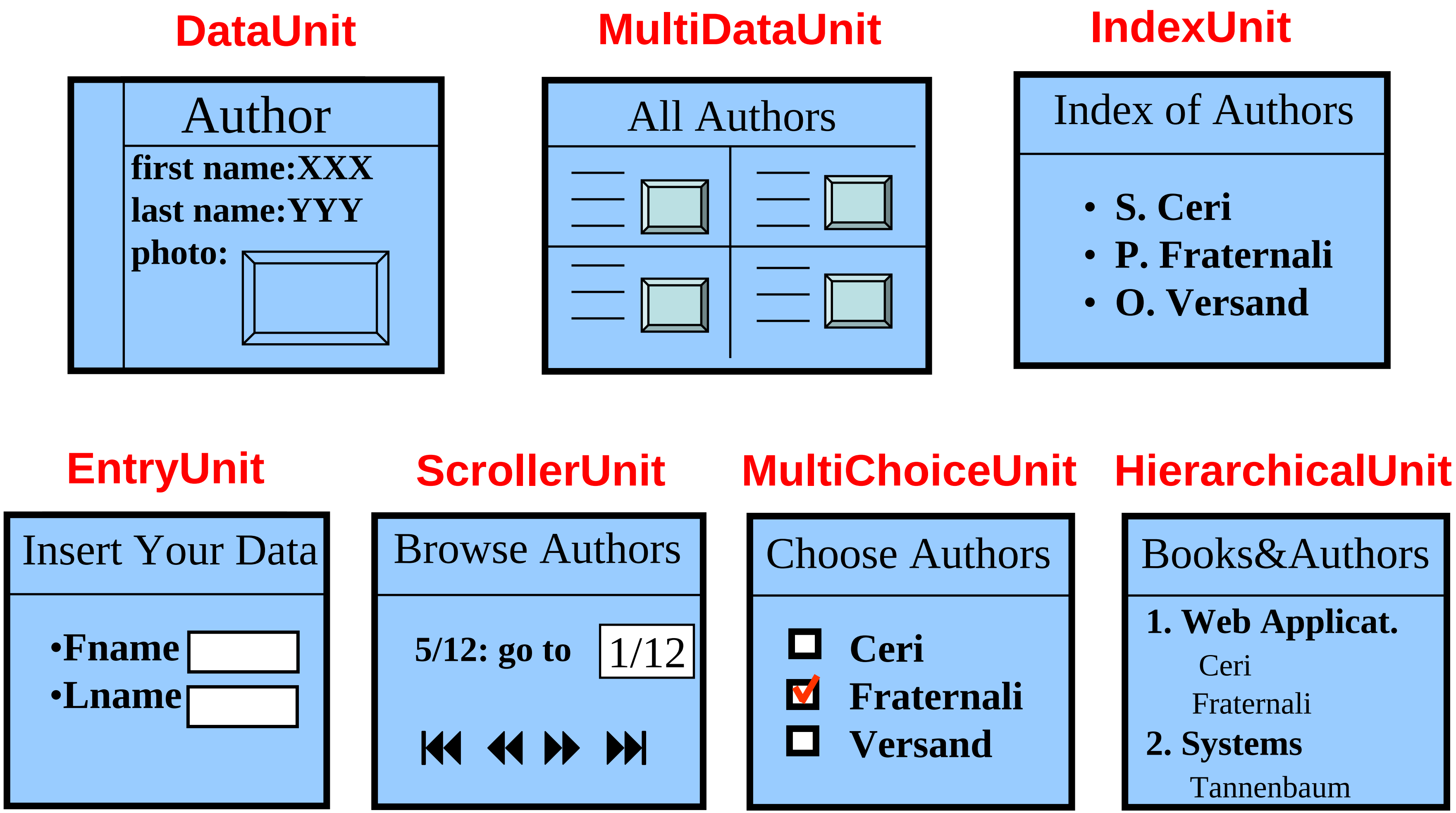
Content Units (2)

- Content
  - instance(s) of an entity
- Selector
  - set of conditions



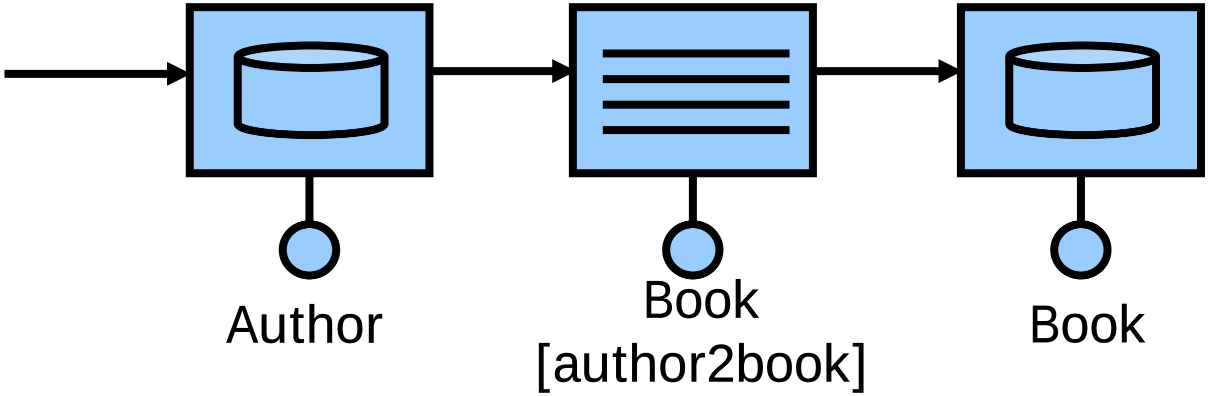


Meaning of Content Units

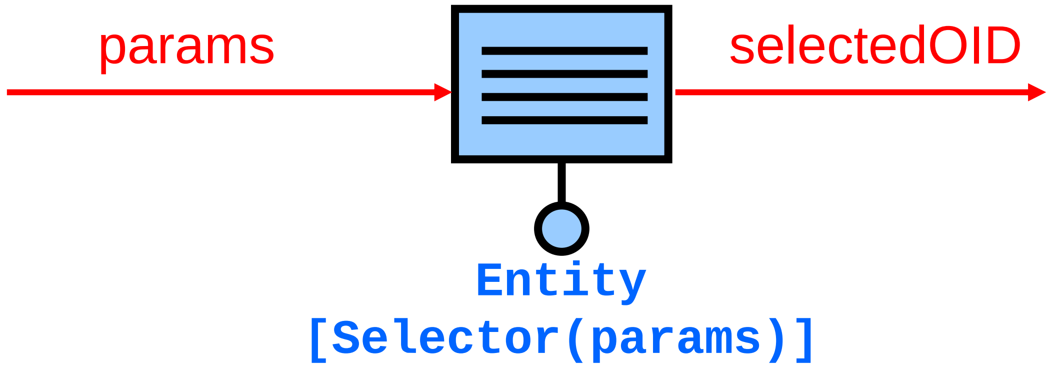


Selectors

- Whenever possible, selectors and their parameters
  - are inferred from the diagram
  - need not be explicitly specified

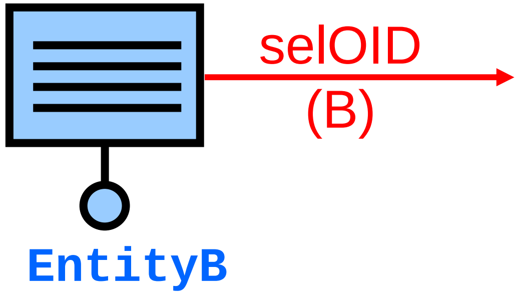


IndexUnit

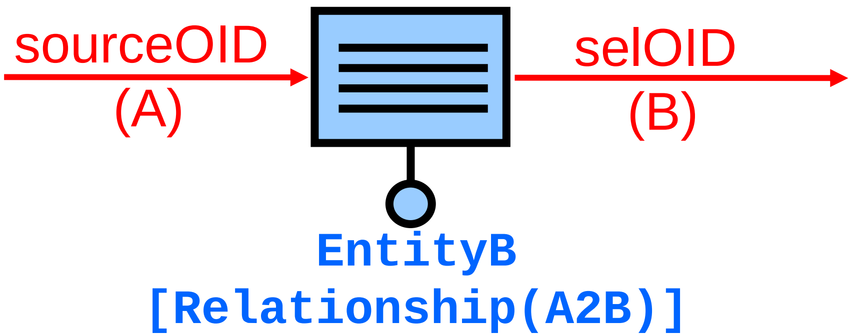


- Index units
  - publish an index of elements (set of objects)
- The container is an entity with (optional) selector
- **Input parameters** for the computation of the selector
  - values for attribute comparisons
  - OID of related object
- **Output parameter**
  - OID of the object selected by the user

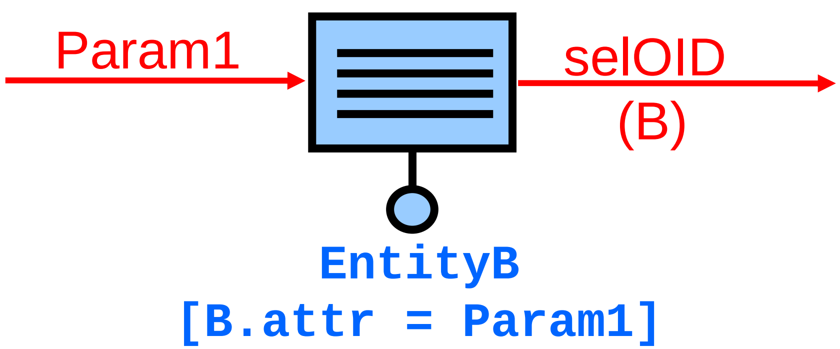
IndexUnit: Examples



- No input links: all objects shown

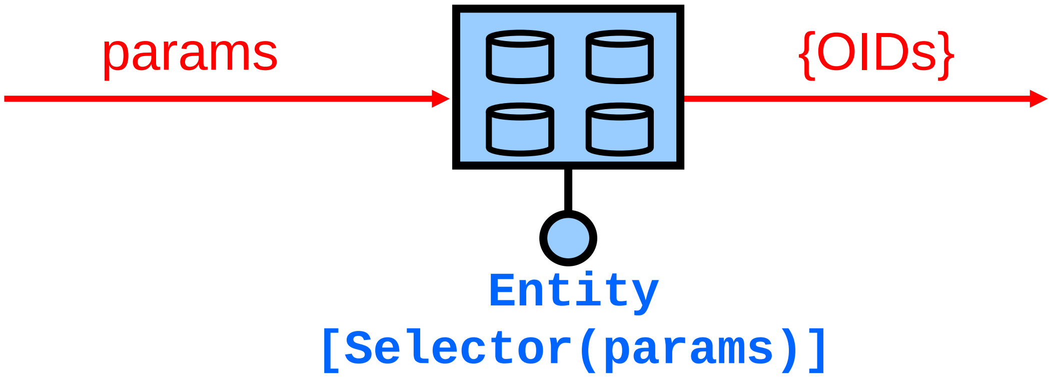


- Input parameter: OID of the source object of the relationship role, all related object are shown



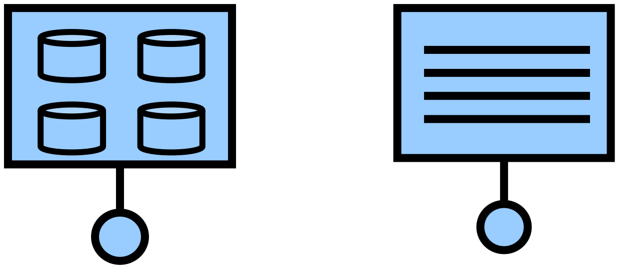
- Input parameter: value to be compared with the attribute, matching objects shown

MultiDataUnit



- MultiDataUnits present multiple instances of an entity (set of objects)
- The container is an entity with (optional) selector
- **Input parameters** requested for the computation of the selector
  - values for attribute comparisons
  - OID of participants to relationships
- **Output parameter**
  - the set of OIDs of the published objects

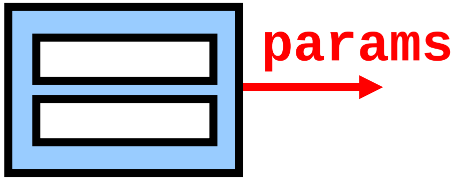
MultiDataUnit vs. IndexUnit



- MultiDataUnits or IndexUnits?
  - index units can be used as access mechanism to shown detailed information about one object
  - multidata units publish information about many objects at time
- The difference is in the output parameter
  - index units output the OID of one selected object
  - multidata units output the OIDs of all the displayed objects

EntryUnit

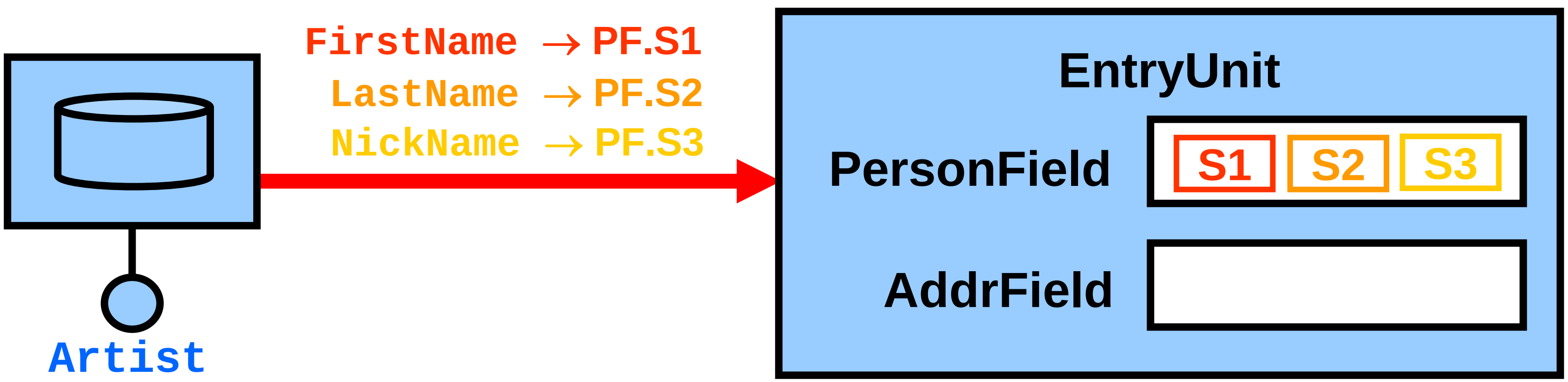
- Unit for describing **input forms** that allow information submission by the user
  - content is shipped to other units via outgoing links (normally one), using link parameters
  - typically translated into HTML using the `<form>` tag and the associated submit button



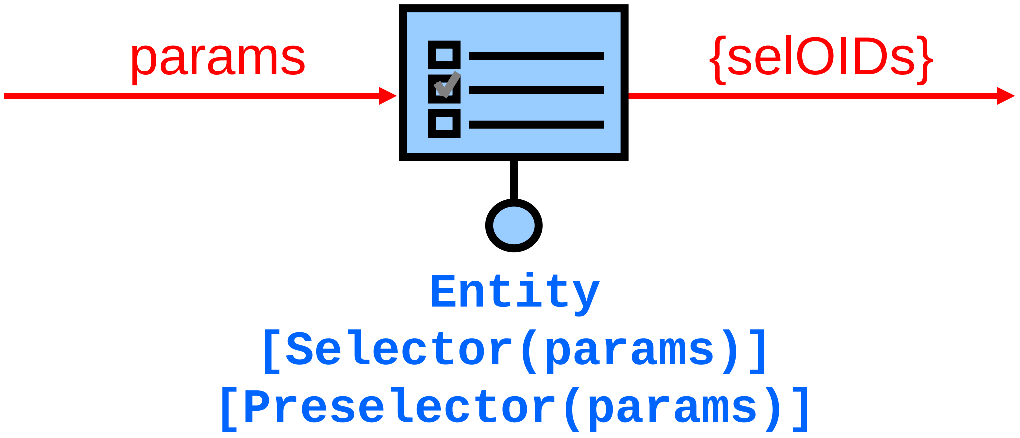
- Entry units contains two types of widgets for data entry
  - **fields** to insert new value
  - **selection fields** to select a value from a list
- Fields have properties
  - preloading (of an existing value in the field)
  - modifiability (Y/N)
  - visibility (hidden/shown)
  - data type (string, integer, text, etc..)
  - input validation rules

EntryUnit: Preloaded Fields

- **A field can be preloaded with value(s)**
  - field slots allow the concatenation of multiple values in the same field
  - slots can contain dynamic values taken from the database or static values defined in the model
- Selection fields need to be preloaded with a set of instances (possibly with multiple slots per instance)
  - the user chooses an instance among them



MultiChoiceUnit



- Publish **index of elements** (set of objects) from which the user selects one or more elements (with checkboxes)
- The container is an entity with (optional) selector and pre-selector
  - **input parameters** requested for the computation of the selectors (values for attribute comparisons and OID of participants to relationships)
  - **output parameters**: OIDs of the objects checked by the user
- **Preselector** allows one to define a sub-set of elements as pre-checked (before any user interaction)

MultiChoiceUnit: Examples

- [ ] Atzeni
- [ ] Ceri
- [ ] Fraternali
- [ ] Versand

- No input links: all objects shown, no pre-checked elements

- [ ] Atzeni
- [x] Ceri
- [x] Fraternali
- [ ] Versand

`[PRE:Relationship(A2B)]`

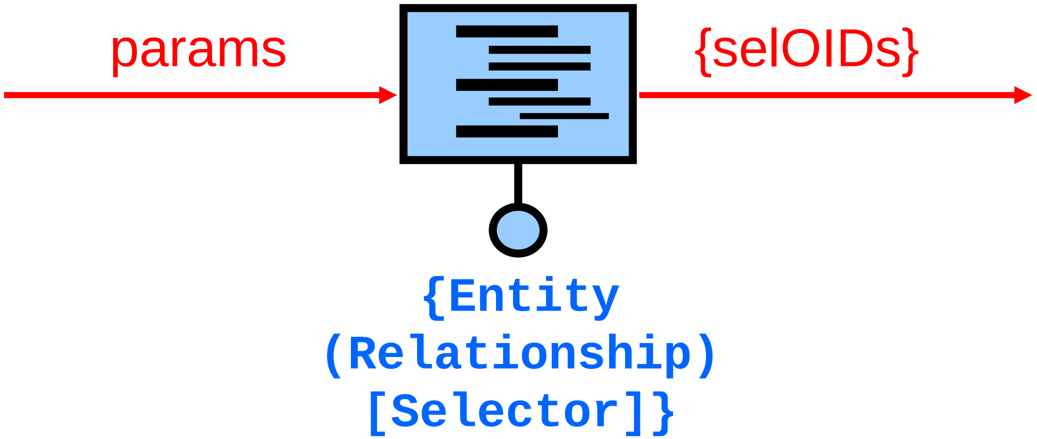
- Pre-selector: all objects shown, pre-checked elements related to A

- [ ] Ceri
- [ ] Fraternali

`[Relationship(A2B)]`

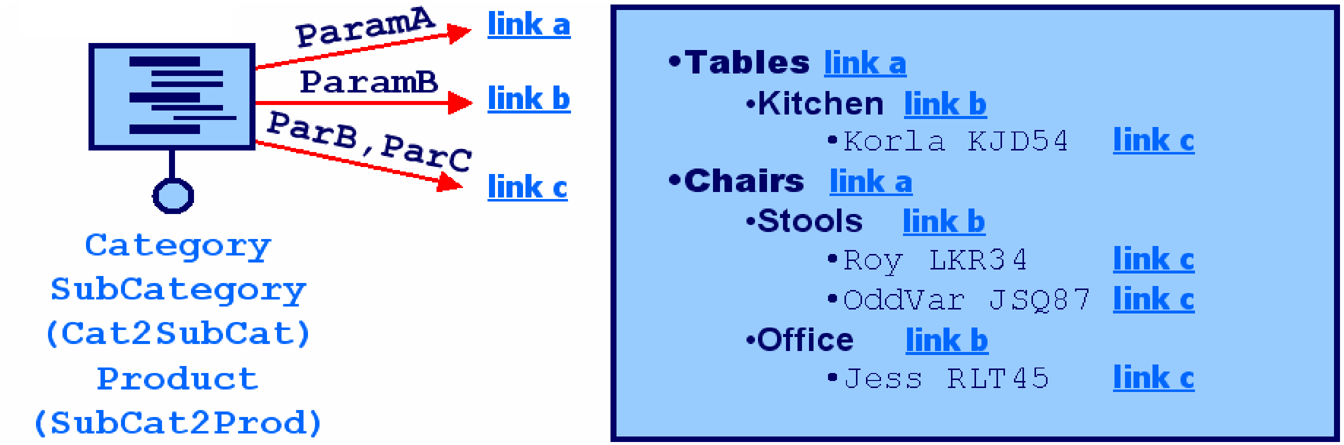
- Selector: only elements related to A shown, no pre-checked elements

HierarchicalUnit

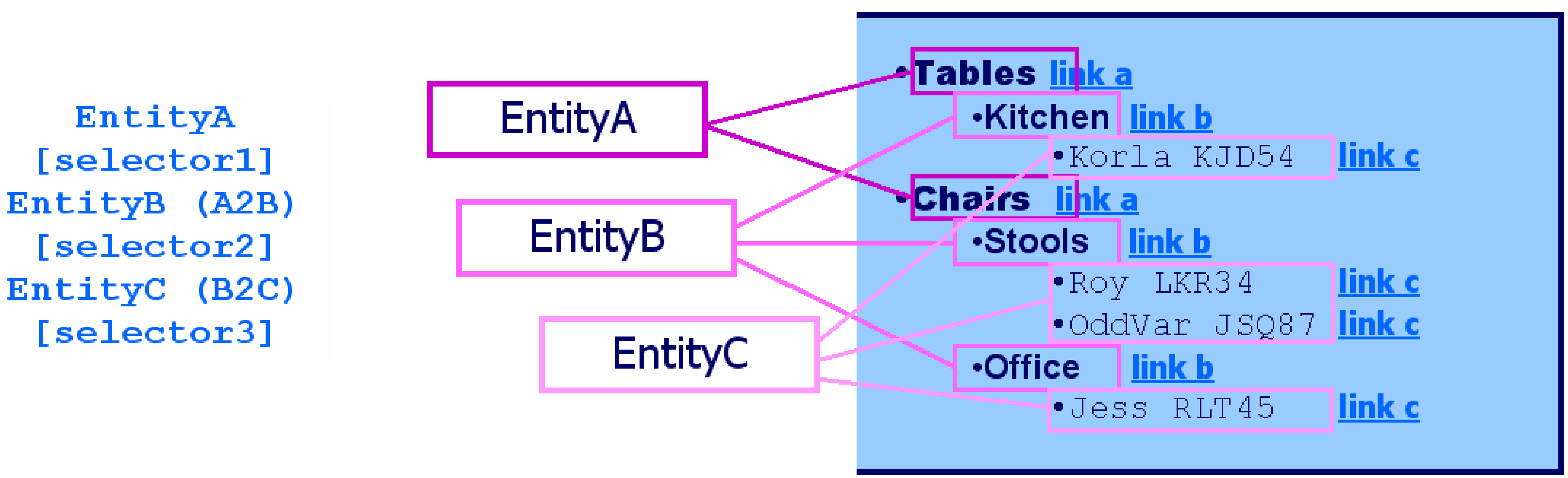


- Publish an **index of elements**, with entries organized hierarchically using entities connected by relationships
- Allow the user to select one element from any level of the hierarchy
- The container is a set of entities and connecting relationships (with optional selectors at all levels)
  - **input parameters** requested for the computation of the selectors
    - values for attribute comparisons
    - OID of participants to relationships
  - **output parameters**: OIDs of the objects selected by the user

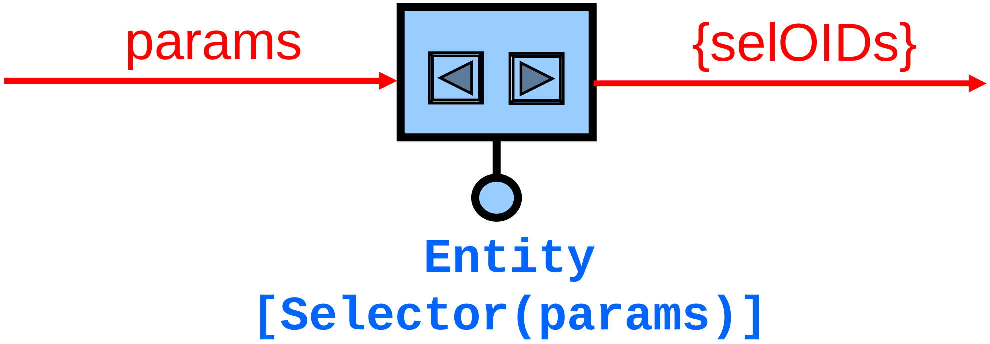
HierarchicalUnit: Outgoing Links



- Each link is rendered as an anchor at the proper level of the hierarchy
- The level where the link is placed depends on the type of the link parameter(s)



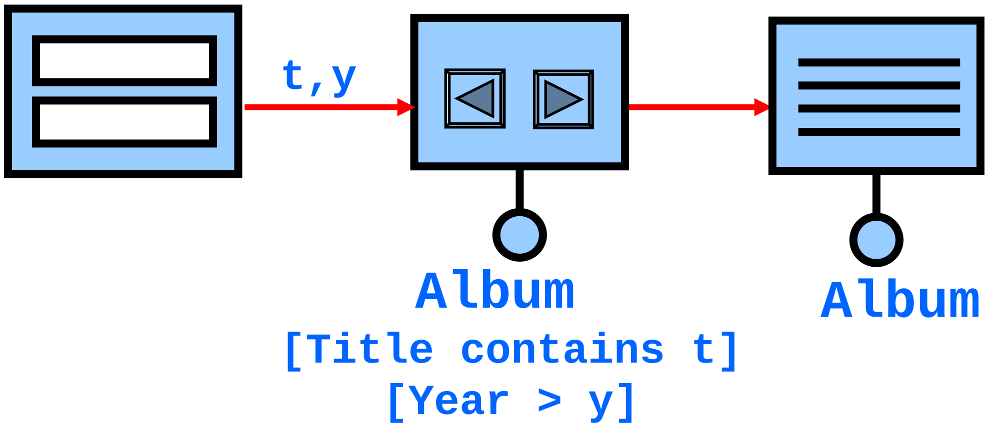
ScrollerUnit

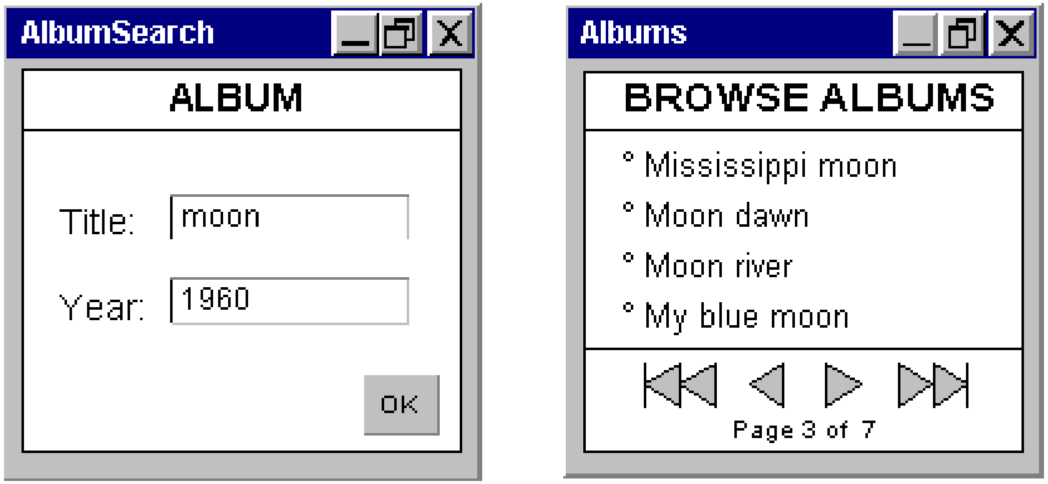


- Add **browsing capabilities** to a set of objects
  - publish links to the first, previous, next, last object of the set
  - used in conjunction with data, index and multidata units
  - number of scrolled objects = block factor
  - **input parameters**: values for attribute comparisons, OID of participants to a relationship
  - **output parameters**: the set of OIDs (possibly 1) of the current block of objects
- Always placed in the same page as another content unit that publishes the current (block of) object(s)

ScrollerUnit + IndexUnit

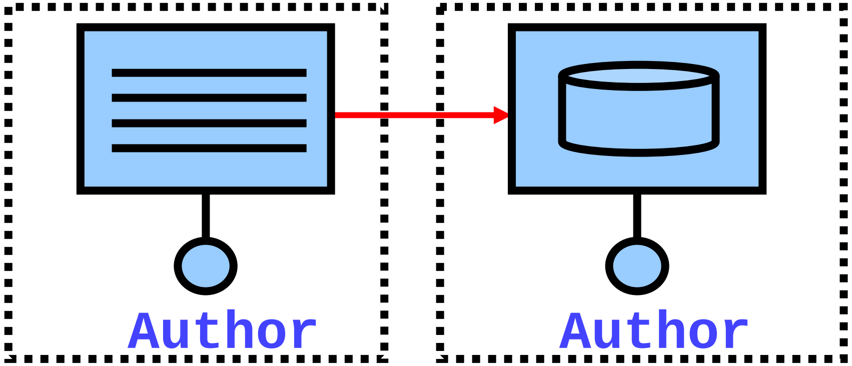
- Paging the result of a search

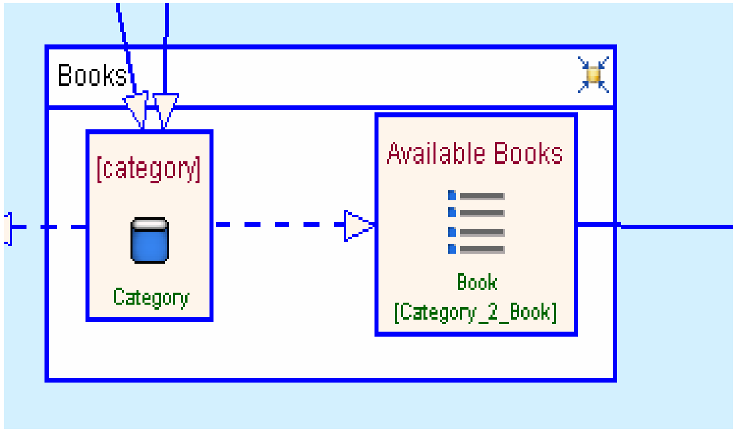




Composition: Pages

- A **Page** is a structured container of units and links
  - possibly structured in and/or sub-pages
  - abstraction of screen, frame, card, deck...
  - permits one to cluster related information for more efficient communication
- Example
  - index of authors and the selected author are shown together in the same page





- **No separation of concerns**
  - hypertext structure and presentation aspects in the same model
  - no distinction between navigation unit (node) and presentation unit (page)

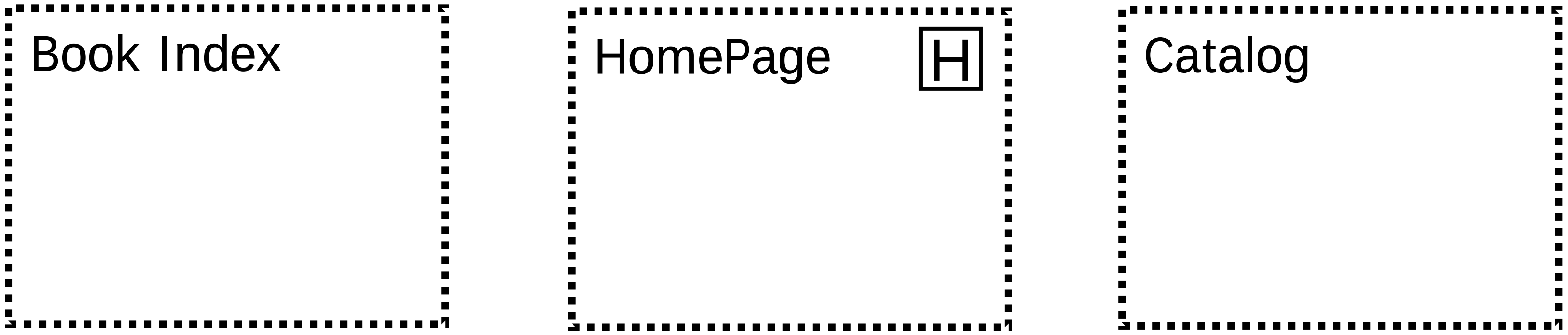
Pages

- **Pages** are the actual interface elements delivered to the user, who browses the hypertext by accessing its pages in the desired sequence.
- A page can also be declared as the **home** of the site view, meaning that it is presented by default when entering the whole site view.
- A page can also be declared **landmark**, meaning that they are globally reachable from all the other pages of the same area or site view,
- A page can be declared **default**, meaning that they are displayed by default when entering their enclosing area.

- WebML also offers the notion of **nested pages**, for modeling the physical organization of some complex pages.
  - nested sub-pages may be in conjunctive form, which means that they are displayed together, or
  - in disjunctive form, which means that the display of one sub-page replaces the display of another sub-page.

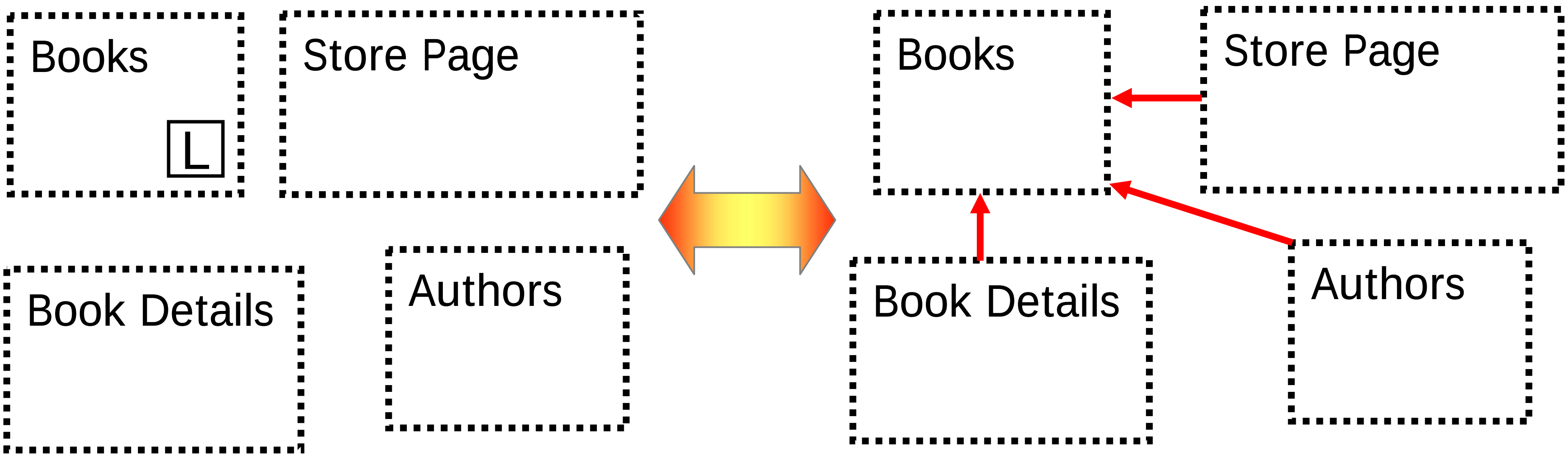
Home Page

- A **page** is a container of one or more pieces of information shown to the user at the same time
  - nesting of pages is allowed: a page can have sub-pages
  - the user navigates a site made of pages
- The **home page** is the main page of a site
  - it is the first page of the site that the user should see
  - each site view must contain a page marked as "Home"



Landmark Pages

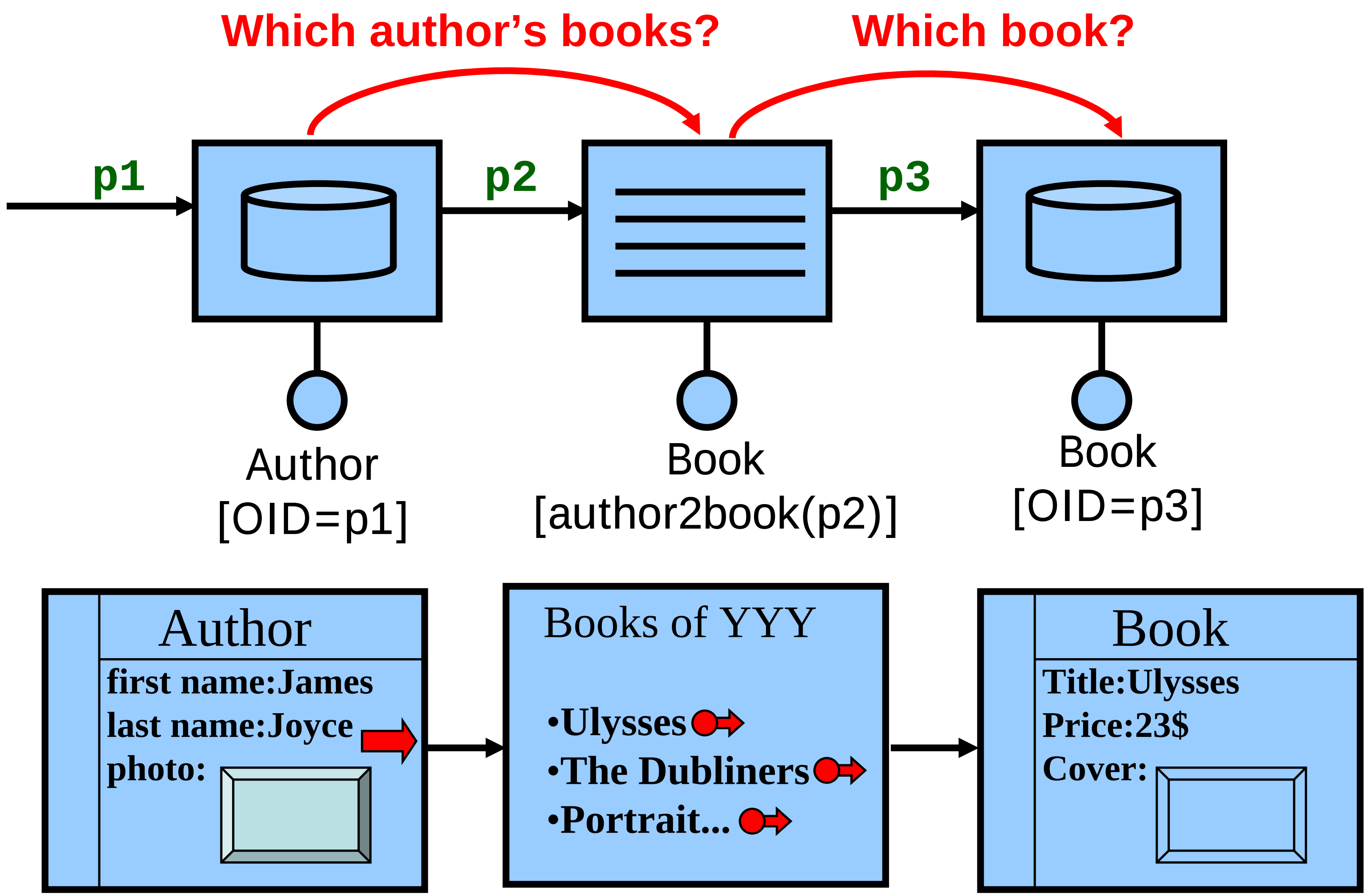
- **Landmark pages** are globally visible pages.
  - the user can jump to them from everywhere in the site view



Links

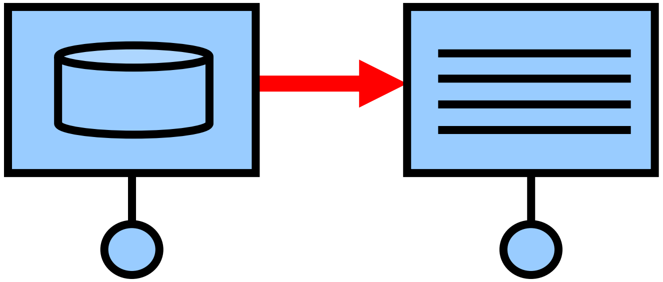
- Semantics of a link
  - moving from one place to another
  - transporting information from one place to another (navigation context)
  - activating a computation (side effect)

Links: Example

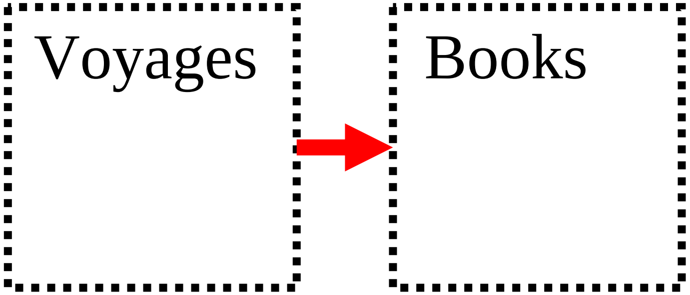


Types of Links

- **Contextual links**
  - directed connection
  - between two units (source unit and target unit)
  - normally rendered by means of anchors or submit buttons
  - context transport

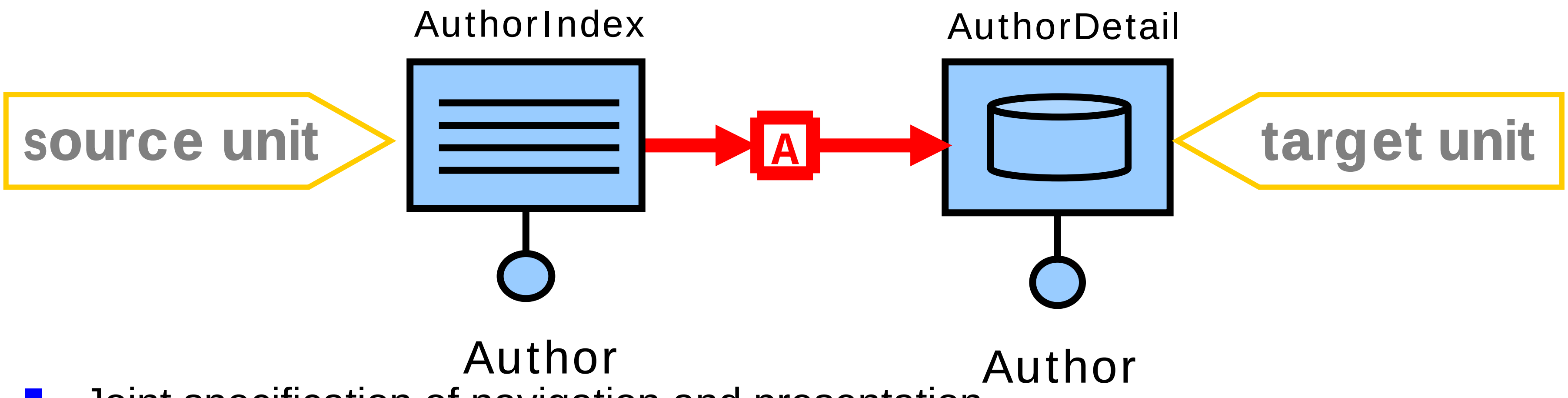


- **Non-contextual links**
  - between pages
  - no context transported
  - the user can browse from a page to another one via an anchor (e.g., *>>Books*)



Automatic Links

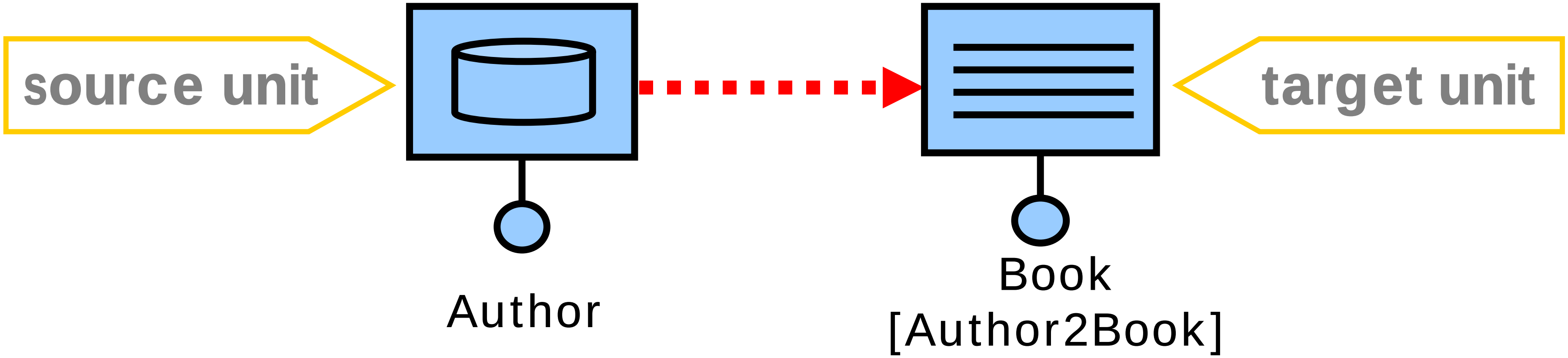
- An **automatic link** passes some default context to the destination unit immediately after the display of the source unit, without the user intervention
- Subsequently, the user can change the passed context by choosing a different object, using the anchor(s) representing the link
- Example
  - together with the AuthorIndex, one object of the AuthorDetail is displayed automatically



- Joint specification of navigation and presentation

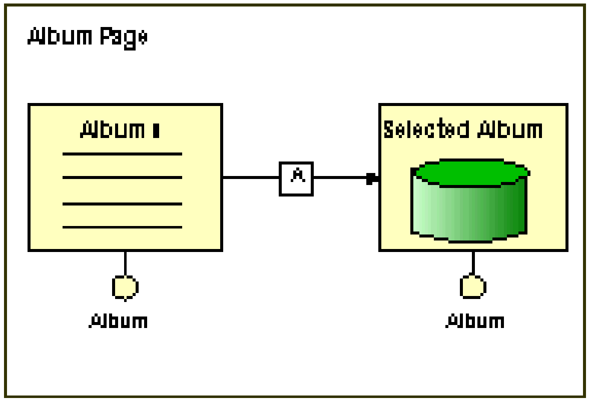
Transport Links

- A **transport link**
  - has a default context that is passed to the target unit
  - immediately after the display of the source unit, without the user intervention
- The user cannot change the default context
- Links are not rendered with an anchor
- Only parameter passing; not user navigation

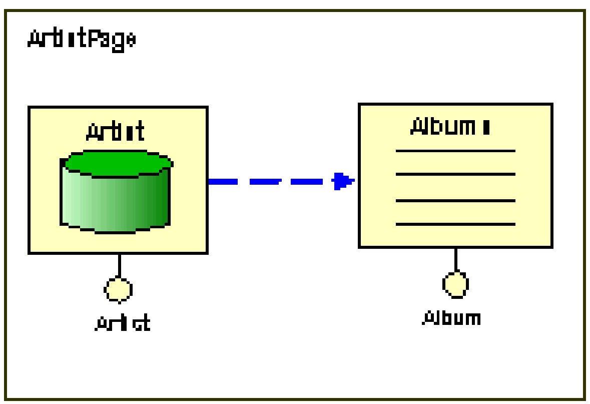


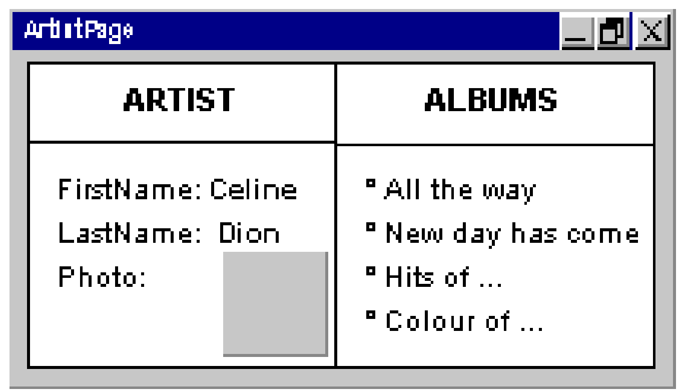
Automatic and Transport Links: Example

Automatic Link Example



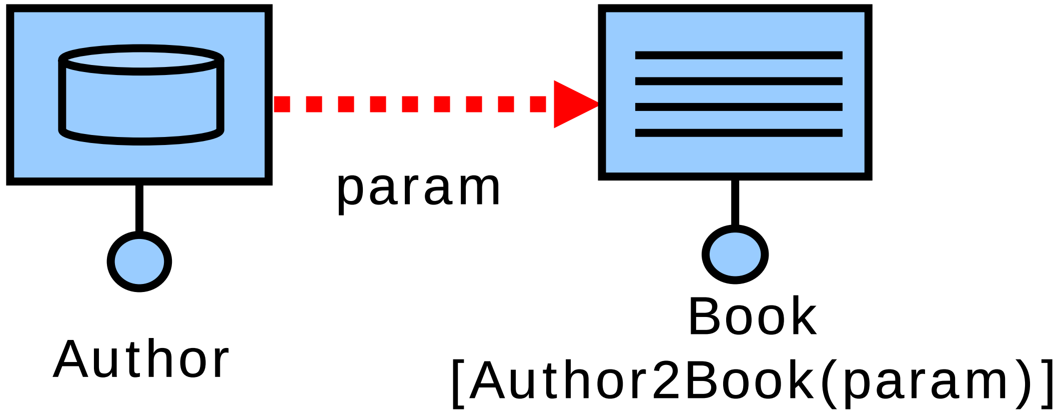
Transport Link Example



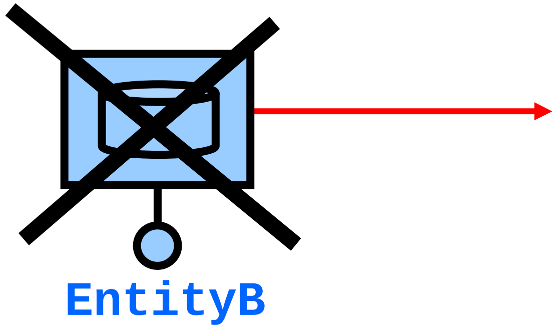


Link Parameters

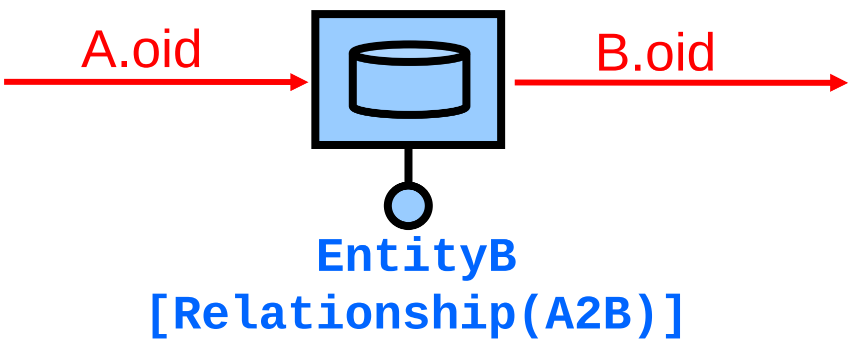
- **Context is transported through links using link parameters**
- A link parameter is defined by
  - name
  - source: one of the output parameters of the source unit
  - target: one of the input parameters of the target unit
- Whenever possible, link parameters are inferred from the diagram and need not be explicitly specified
  - diagrams become simpler and more readable



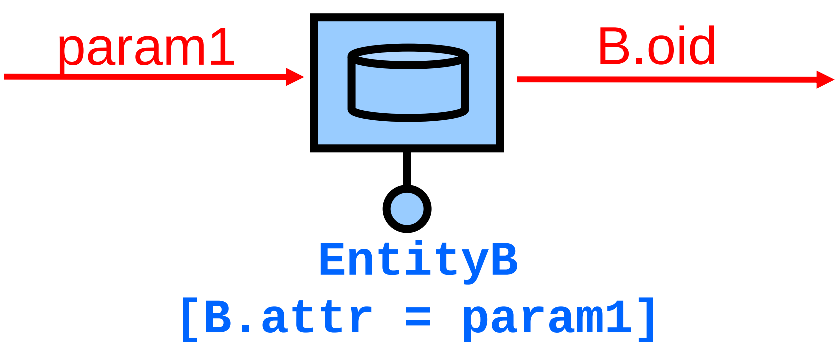
Link Parameters: Examples



- No input links and no selector: wrong unit!



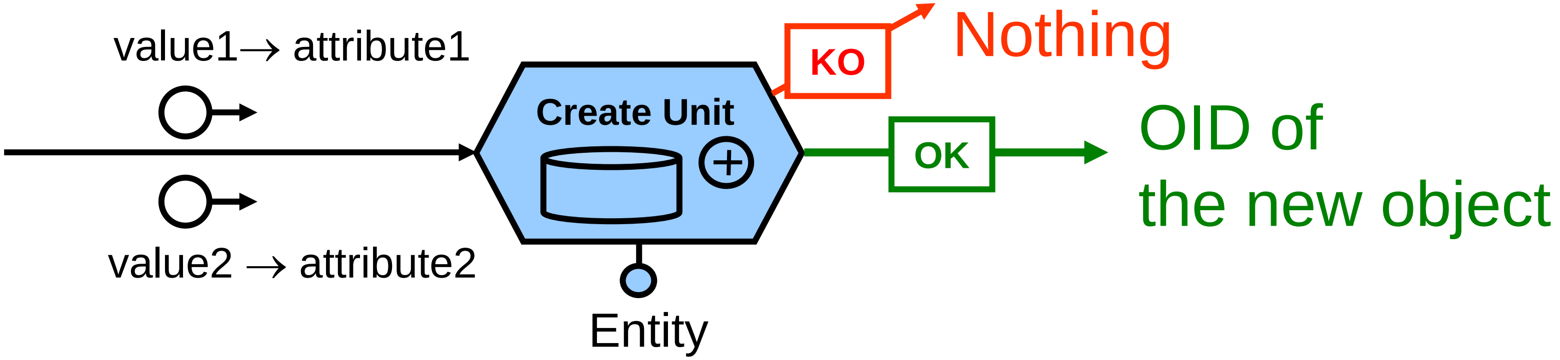
- Input parameter: OID of a connected object: the related object is shown



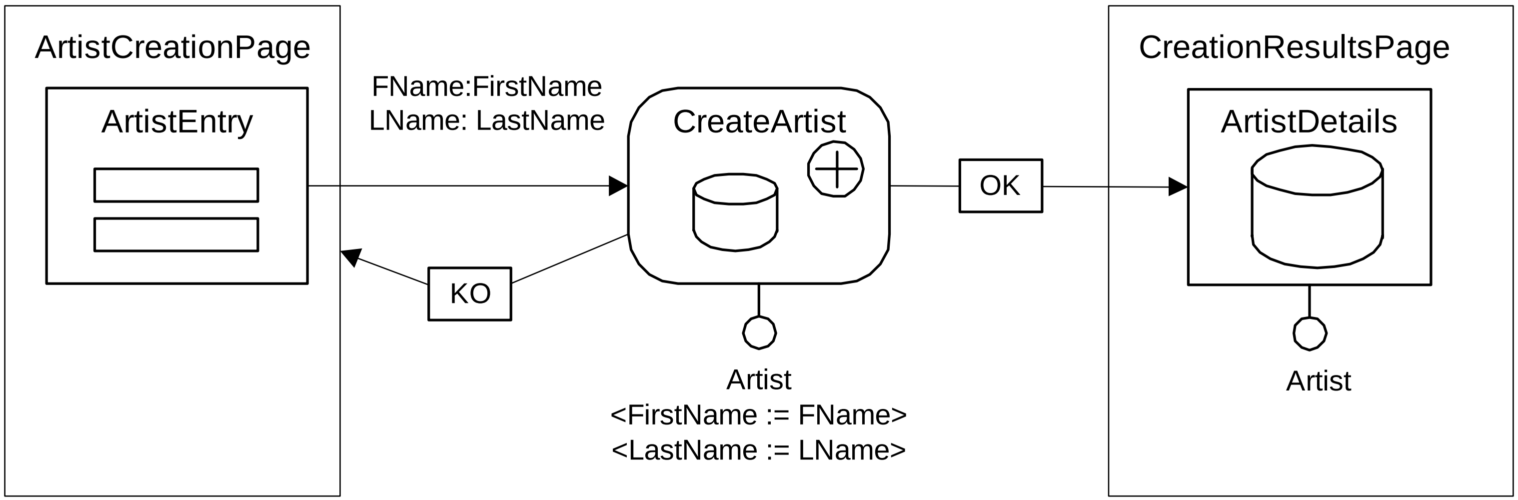
- Input parameter: value to be used in the selector: the matching object is shown

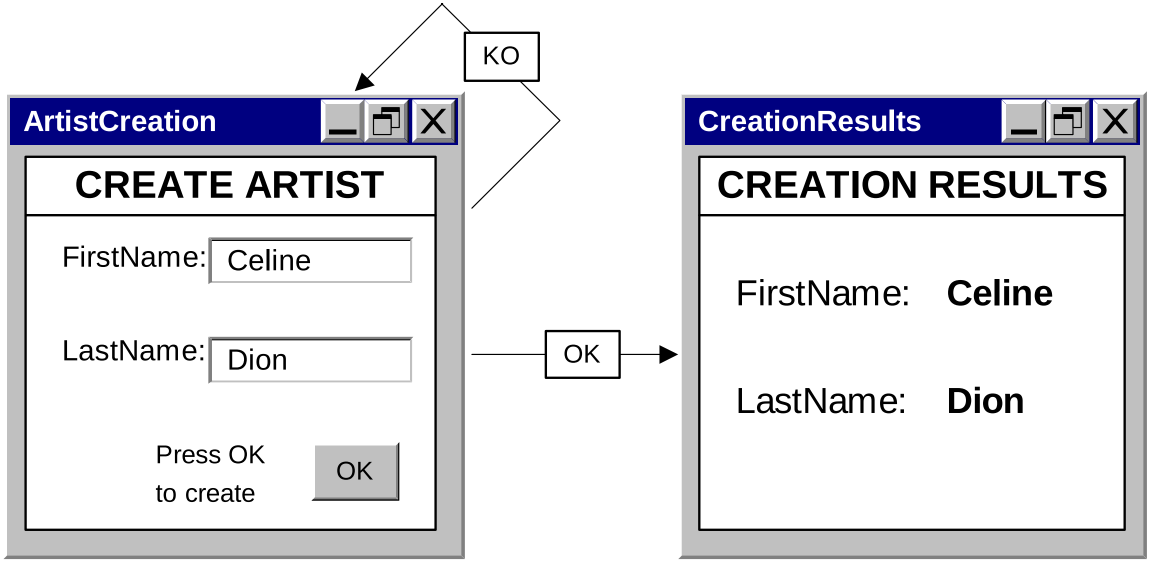
Operations

- **Operation units (operations, for short) model arbitrary actions that can be triggered during the navigation of an hypertext.**
- Operations may accept input parameters and emit output paramaters, have input and output links.
- Operations do not display information, and thus are placed outside pages.
- To trap the occurrence of failures, WebML distinguishes two types of output links thay may exit from an operation
  - OK links: are followed after the successful exectuion of the operation
  - KO links: are followed when the execution of the operation fails



Operations: Example

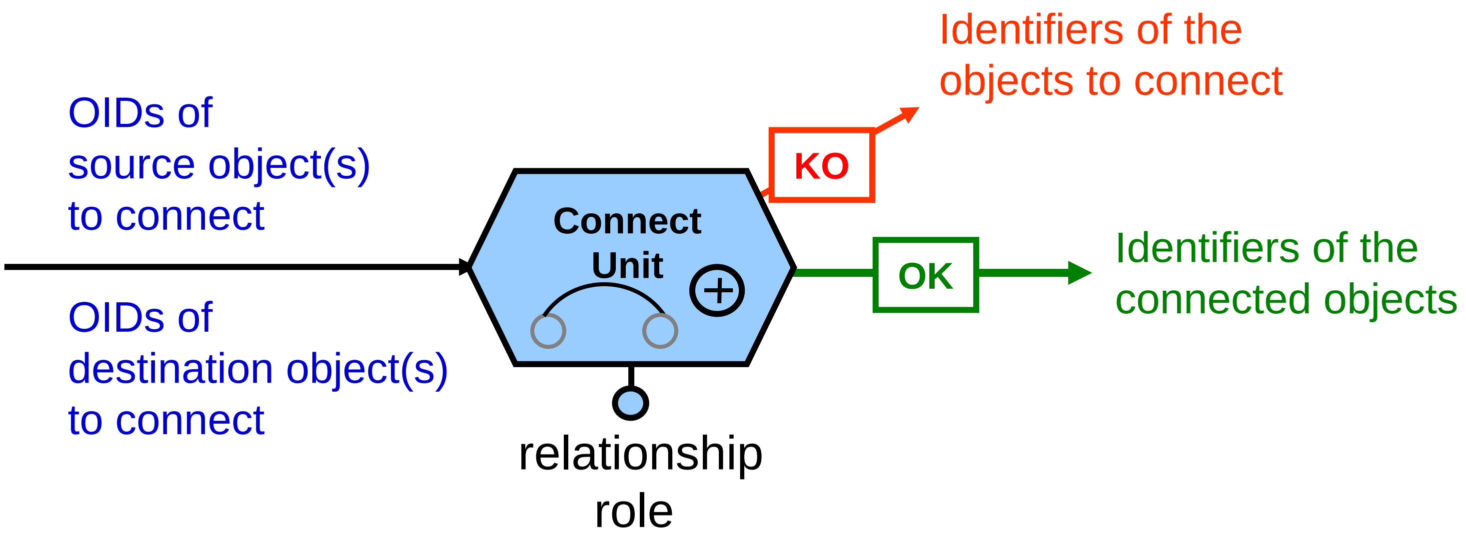




Predefined Operations

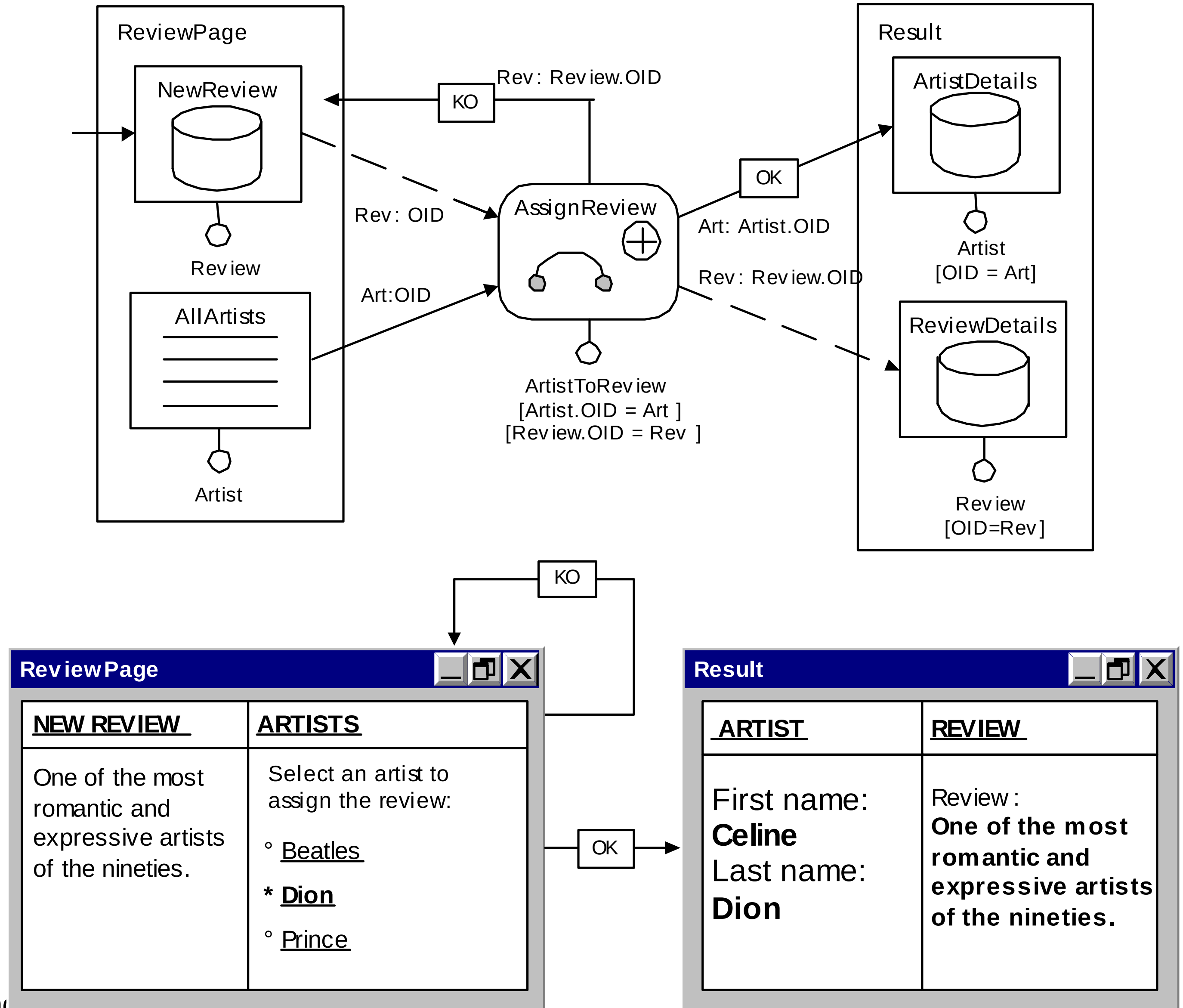
- **WebML provides a few predefined operations**
  - for content management (create, delete, modify, connect, disconnect)
  - for access control (login, logout)
  - for communication (sendmail)
  - developers are free to invent other operations, as far as they understand their input, business logic, and output.

Operation Connect

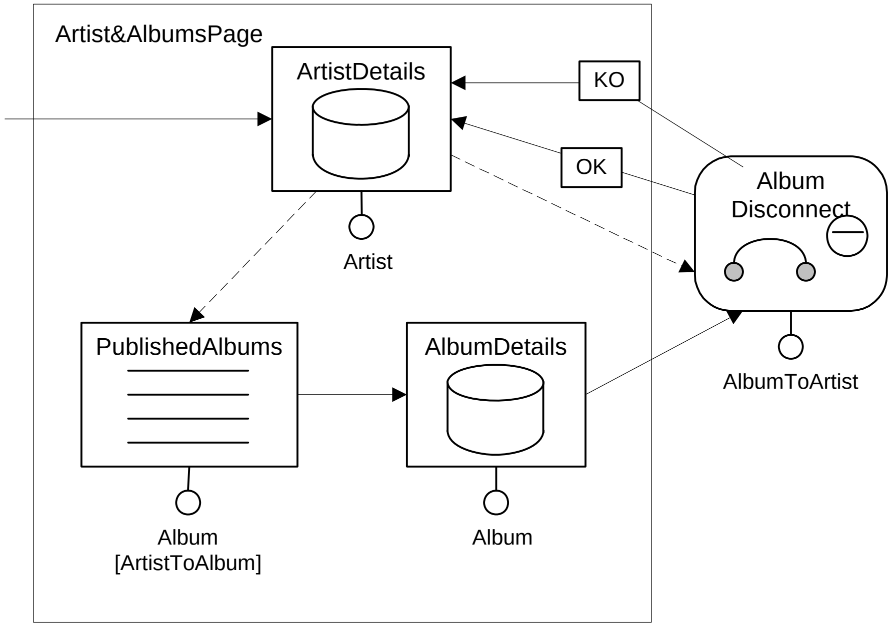


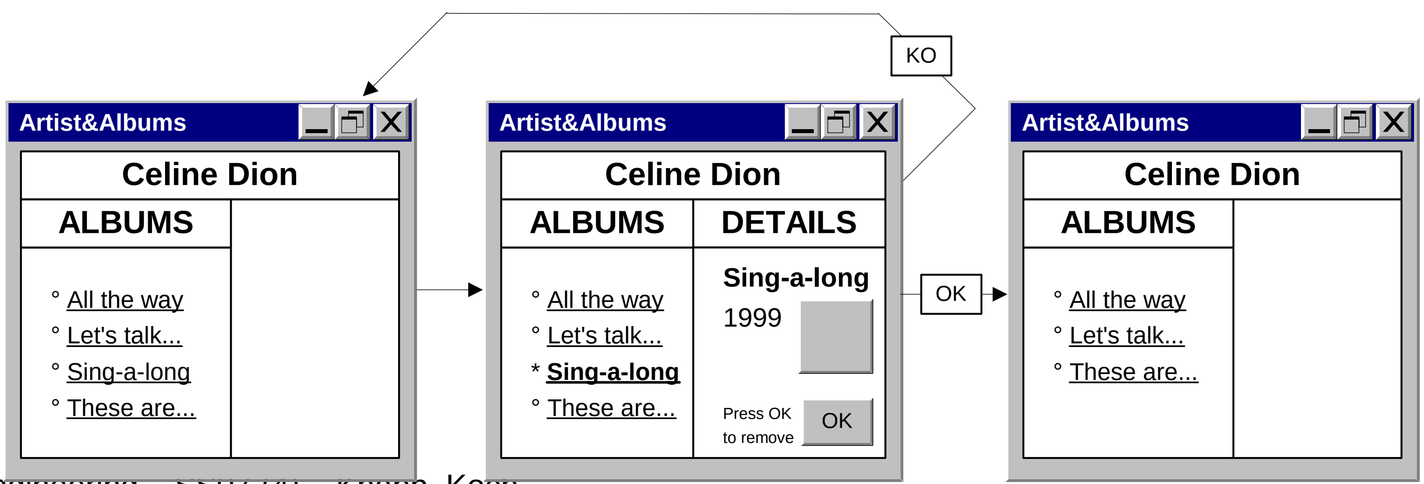
- If the any of the input parameters is a set of OIDs, the operation is applied to each element of the set
- The **disconnect operation** is similar

Operation Connect: Example



Operation Disconnect: Example





Areas

- **An area is a set of logically homogeneous pages**
- Examples of areas
  - sections of a portal: sport, music, technology, …
  - elements of a data-management system: products management, news management, …
- Areas can be nested, so that sub-areas can be defined inside areas
- Each area should have a default page or a default sub-area, to give a meaning to landmark areas and non-contextual links pointing to areas

Site Views

- **A site view is a set of pages and/or areas forming a coherent view of the site**
- Multiple site views can be defined on the same data model
- Different site views can be published for different types of users and for different types of output devices
- Site views can be
  - public: everyone can enter
  - private: access control with password protection is enforced

Role of XML in WebML Models

- All information of your modeled applications can be stored in XML files
- A DTD is defined for WebML schemas XML files
- Automatic generation of Web pages in the selected publishing language can be done by applying XSL transformations

```
<!-- ================================================================ -->
<!--  INDEXUNIT                                                       -->
<!-- ================================================================ -->
<!ELEMENT INDEXUNIT (SELECTOR?, DISPLAYATTRIBUTE*, SORTATTRIBUTE*,
                      LINK*, PROPERTY*, COMMENT?)>
<!ATTLIST INDEXUNIT
          id                  ID                #REQUIRED
          name                CDATA             #IMPLIED
          entity              IDREF             #IMPLIED
          distinct            (yes|no)          'no'
>
```

Presentation

- **Presentation dealt with along two lines**
  - gallery of default presentation styles applicable to site views or single pages (with elementary unit positioning on a grid)
    - each presentation style is an XSL style sheet
  - compatibility with best selling tools for presentation editing, for advanced page design
    - e.g., WebML extensions for Dreamweaver 3 & 4

Presentation Model

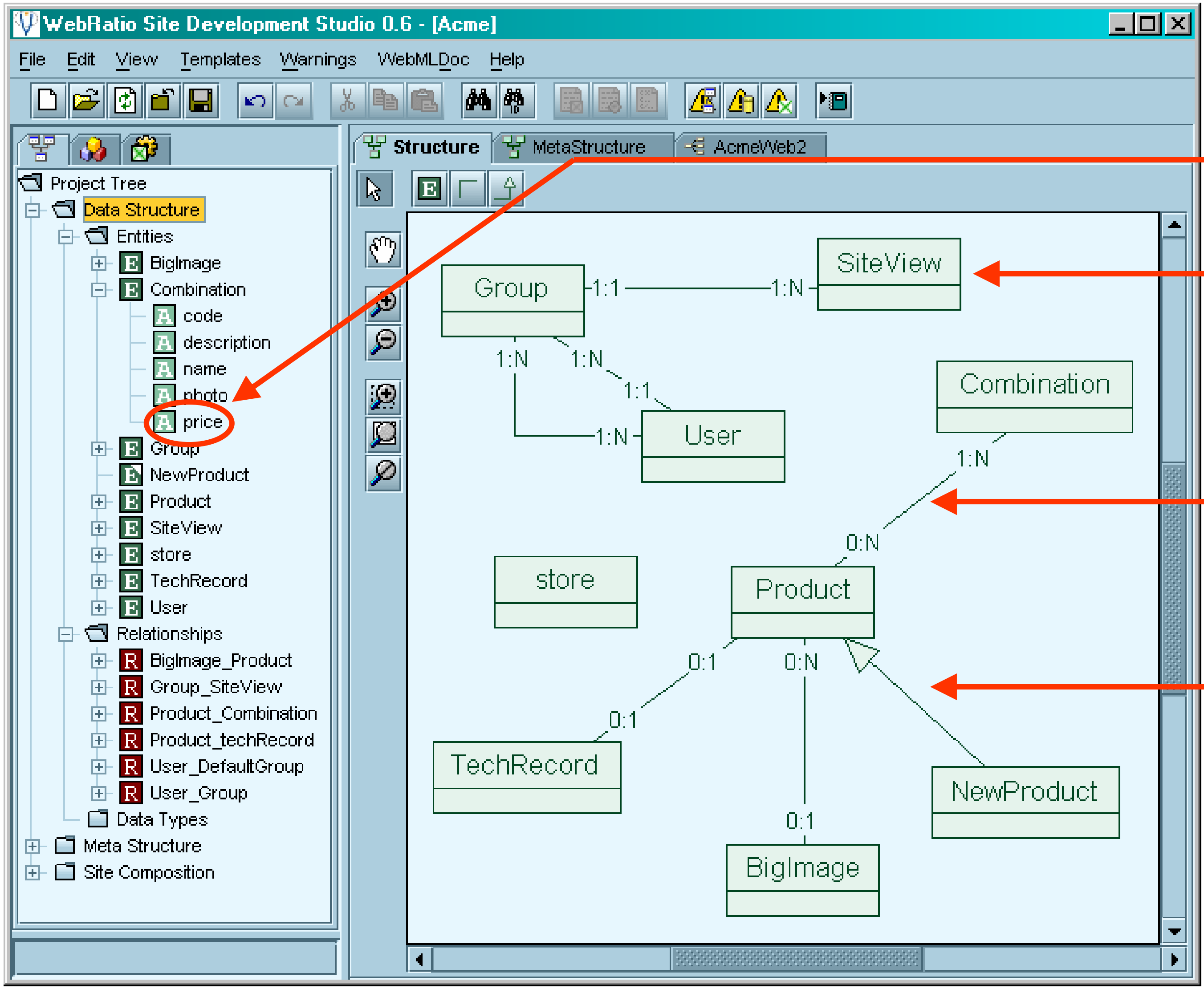
- **Presentation is the task of defining the look and feel of pages**
- WebML does not include a specific model for expressing presentation at the conceptual level
- Since WebML specifications can be represented using XML, presentation is considered like a document transformation mapping the WebML specification of a page into a page written in a concrete implementation language like JSP or ASP.NET.
- Presentation is addressed in WebML by attaching **XSL style sheets** to site views, pages, units and unit subelements.
  - XSL style sheets take in
    - input WebML specifications, coded as XML documents conforming to the WebML Document Type Definition, and
    - output page templates embodying the required mark-up code and data access queries.
  - an implementation of WebML may include several pre-defined presentation style sheets and the server-side components supporting the data access queries needed to populate the content of the page templates produced by the XSL style sheets.

WebRatio Site Development Studio

- **Structure model design**
  - data derivation (derivation wizard)
- **Hypertext model design** (siteviews)

- Consistency checks (warnings)
- Structure Mapping onto a datasource
- Elementary units positioning in the pages
- Automatic web site generation with (default) presentation styles

WebRatio: Structure Modelling



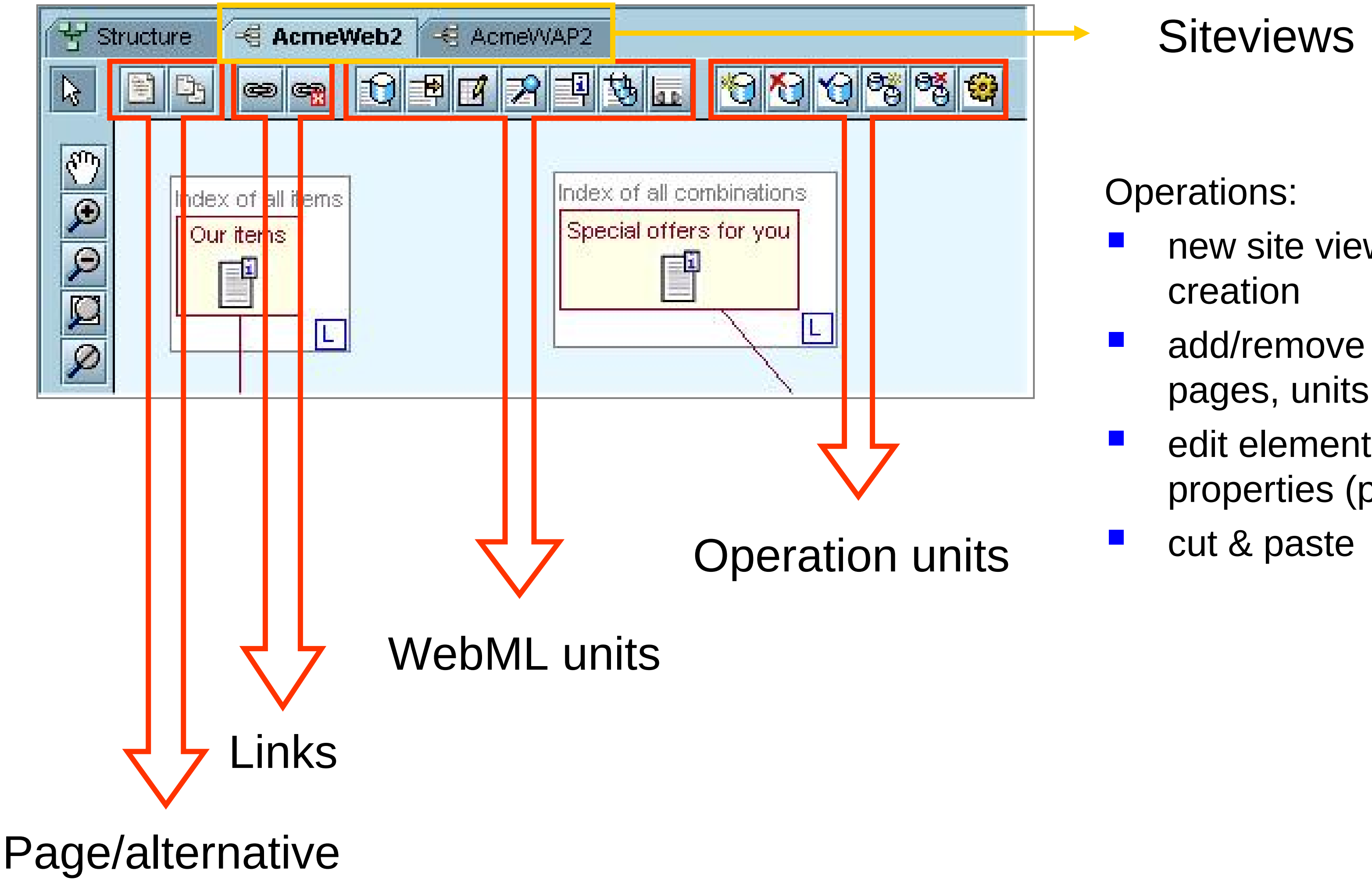
Attribute

Entity

Relationship

ISA hierarchy

WebRatio: Hypertext Modelling



Operations:

- new site view creation
- add/remove pages, units, links
- edit elements properties (panel)
- cut & paste

Literature

- Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, Maristella Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufman Publishers, Elsevier Science, 2003.
- WebML. `http://www.webml.org`
- WebRATIO. `http://www.webratio.com`