Conceptual Modeling of Data-Intensive Web Applications

Breaking Web applications into "skeletons" that describe their core, access, and connection schemes can help designers understand existing sites and engineer effective new ones.

Stefano Ceri,
Piero Fraternali,
and Maristella Matera
Politecnico di Milano

Many of the Web applications around us are *data-intensive*; their main purpose is to present a large amount of data to their users. Most online trading and e-commerce sites fall into this category, as do digital libraries and institutional sites describing private and public organizations. Several commercial Web development systems aid rapid creation of data-intensive applications by supporting semiautomatic data resource publishing.[1] Automatic publishing is typically subject to the constraints of database schemas, which limit an application designer's choices. Thus, Web application development often requires adaptation through programming, and programs end up intricately mixing data, navigation, and presentation semantics. Presentation is often a facade for elements of structure, composition, and navigation.

Despite this frequently unstructured development process, data-intensive applications — based on large data sets organized within a repository or database — generally follow some typical patterns and rules. In this article, we will describe these patterns and rules using WebML as a conceptual tool to make such notions explicit.[2] WebML is a conceptual Web modeling language that uses the entity-relationship (ER) model for describing data structures and an original, high-level notation for representing Web content composition and navigation in hypertext form.

Using WebML, we can abstract data-intensive Web sites as complex arrangements of elementary structures called *skeletons*, which are pairs of data and hypertext diagrams. Identifying skeletons can improve our understanding of existing Web applications and guide our design of new sites through the reuse of effective application components.[3] Skeletons also facilitate site wrapping to let us integrate services and information extracted from existing sites.[4] We introduce some standard skeletons that have been identified, and we illustrate them with examples from real Web sites.

WebML Overview

The WebML modeling language is one of several proposals that have emerged from recent research on conceptual models and tools for the Web. Fraternali has published an overview and comparison of much of this work.[1] All the approaches focus on conceptually modeling Web applications as a way to generate specifications that allow reasoning at a high level of abstraction without committing to detailed architectural and implementation issues. WebML's distinguishing characteristic is an extensive use of diagrams supported by a powerful GUI.

WebML provides modeling abstractions that a computer-aided software engineering (CASE) environment can translate into concrete page templates. According to the WebML approach, Web applications have two orthogonal conceptual dimensions:

- The *data model* describes the schema of data resources according to the ER model.
- The *hypertext model* describes how data resources are assembled into information units and pages, and how such units and pages interconnect to constitute a hypertext.

Let's look at how we can use WebML elements to express pages from a real Web site. (See webml.org for a more complete and formal definition of WebML.[2])

Data Design

To create a Web application we start with the data design phase, during which the designer specifies the data organization in terms of the relevant entities and relationships. WebML supports the classical ER model with generalization hierarchies, typed attributes, and cardinality constraints. Figure 1 shows a data schema for the data underlying a book sales site inspired by Amazon (www.amazon.com). The schema groups books into categories and shows that a book can be available in several editions. Each book is also connected to other books bought by users who have purchased it; the site can use this relationship to give a buyer additional recommendations. Users order a specific edition of a book by filling an order line with information about the book edition purchase, such as the quantity of ordered items; the shopping cart associated with a given user groups all the order lines of a purchase.

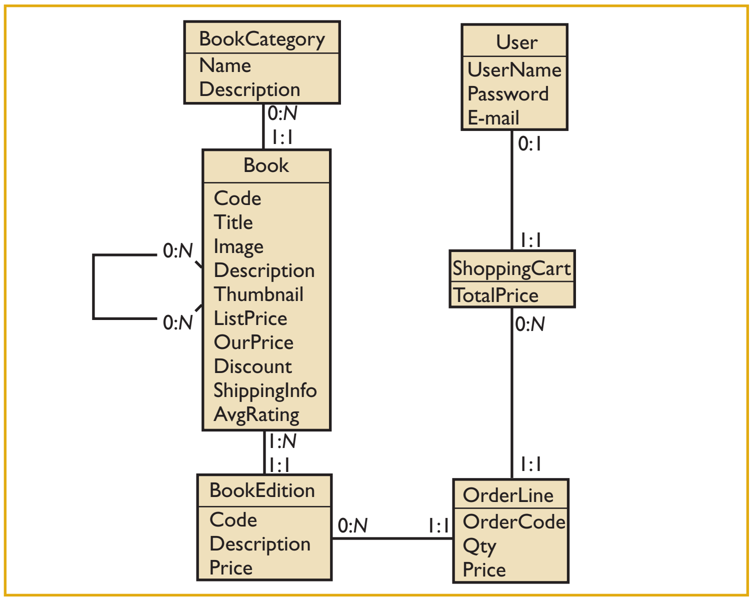


Figure 1. Data schema example, describing the content organization of a book sales site inspired by Amazon. The Web site groups books by category, includes information about each edition of a particular title, and tracks orders inserted in the user's shopping cart.

Hypertext Design

The application development process next proceeds with hypertext design, through which the designer defines the Web site's hypertext topology — the organization of content by basic units, links between them, and unit composition within pages. Table 1 (next page) lists the five types of content units used in hypertext design. The first four units are for displaying one or more instances of the data schema entities, selected by querying entity attributes or relationships. The fifth unit, data entry, is for collecting input values through forms.

Links relate units and express Web site navigation as well as information transfers from one unit to another. Some links also trigger computations, such as content update operations activated by a `submit` link that users follow after entering new data. The designer composes content units into pages, which represent abstractions of self-contained portions of content treated as independent interface blocks delivered to the user (as with HTML pages).

Figure 2 (next page) shows a hypertext fragment inspired by the Amazon Web site, illustrating WebML's expressive power for abstracting and conceptualizing a Web site's access mechanisms. The Home page provides access to books through two alternative mechanisms: an index of the available book categories (`Categories` index) and a form (`BookSearch` entry) for starting a key-based search of books.

If the user chooses the index, the site presents

Table 1. Content units of the WebML composition model.

Unit	Visual notation	Description
Data	Data unit	Shows data about a single entity instance.
Multidata	Multidata unit	Shows data about several entity instances.
Index	Index unit	Shows a list of properties (also called descriptive keys) of a given set of entity instances. A user clicks on an index entry to select and display one instance.
Scroller	Scroller unit	Provides commands for scrolling through objects in a list — for example, the sequence of all the instances of an entity. Scrolling commands let the user move to a set's first, last, previous, and next elements and select and display one instance of the set.
Data entry	Entry unit	Shows a form with several fields for collecting user input. Input might be conditions used for searches over entity instances or parameters for operations such as content updates, logins, and generic external operations.

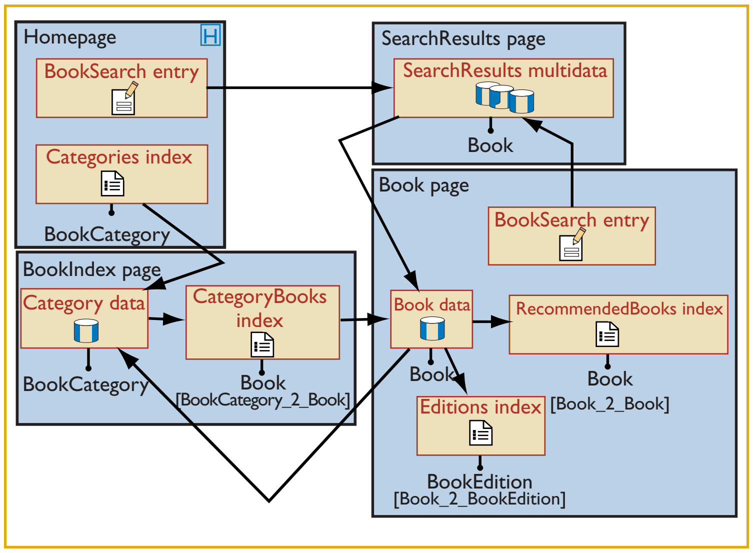


Figure 2. Hypertext design. This simplified hypertext specifies a browsing path through the catalogue on a book sales Web site.

the `BookIndex` page, which includes data about the selected category (`Category` data) and the index of its books. To perform a search instead, the user enters search criteria in the `BookSearch` entry unit, and the `SearchResults` page uses a multidata unit to display information about books matching the criteria. Both access methods eventually yield a specific `Book` page, which shows information about that book (`Book` data) together with an index of all editions of it (`Editions` index) and the `RecommendedBooks` index based on purchase history. The `BookSearch` entry unit lets users specify a search predicate; this entry unit is the same as that included on the Home page. Readers can recognize several of the preceding conceptual elements in actual Amazon Web pages.

Write Access and Content Management Operations

WebML also includes operation units we can use to invoke external operations that manage and update content. These operations can, for example, create, delete, or modify an entity instance, or they can add or drop a relationship between two instances.[5] Table 2 lists each operation unit with its visual notation and a brief description.

Figure 3 shows a simplified hypertext fragment in WebML that models the creation of order lines, their insertion into the user's cart, and their deletion and modification. From the `Book` page, a user selects the edition (for example, hardcover or paperback) and then goes to the `NewOrderLine` page to enter book purchase information such as order quantity. Once data entry is complete, a chain of *create-and-connect* operations generates an order line and joins it to the book edition and the user's cart. The site then displays the `ShoppingCart` page, which contains the accumulated information, including the user's name (`User` data), the cart total price (`ShoppingCart` data), and the order lines (`Orders` multidata). The user can modify an order line's quantity (via the chain of `QuantityEdit` entry and `ModifyOrder` units) or delete order lines (via the `DeleteOrder` unit).

Recognizing Basic Information Concepts

An important step in developing skeletons is classifying the roles that concepts can play in the data schema. We distinguish three types of concepts:

- *Core* concepts are central to the application, constituting the application's main asset and expressing its mission. Examples include a product sold by an e-commerce site and a personal message published on a community Web board. Core concepts form the Web application's backbone.
- *Access* concepts support the location of core

concepts. They superimpose a categorization that can be used to express index hierarchies, which progressively lead users from the Homepage to the core content, or to express search mechanisms focused on well-defined classes of core concepts. Access concepts can also define subsets of representative core concepts as meaningful collections (such as “pick of the day” or “site’s best choices”).

- *Interconnection* concepts connect core concepts and are typically expressed through relationships that users can navigate to move the focus from one core concept to a related one. In some cases, entities can also represent interconnection concepts.

This type of reasoning is useful for abstracting a Web site’s information content and decomposing its ER schema into well-identified subschemas:

- The core subschema collects entities and relationships denoting core concepts.
- The access subschema collects entities and relationships playing the role of access facilitators.
- The interconnection subschema includes the entities and relationships that connect core entities.

These classifications are our starting point for identifying WebML skeletons — pairs of ER subschemas and WebML hypertext diagrams that describe structures that recur within Web applications.

Our skeletons present simplifications of real cases, in which pages frequently include extra information such as advertising and ornamental HTML markups. You can, of course, use WebML to model these elements, but our skeletons ignore them because they are not relevant for understanding the structure of content and navigation. In addition, real Web applications often cluster a lot of information within pages, possibly collapsing many units within a single page. Our skeletons feature simpler compositions that highlight pages’ most significant content units. Our skeletons are easily adaptable to more complex compositions, but keeping them simple brings to light the regular structure of many Web applications. This improves our understanding of them and consequently helps us with their design, reverse engineering, and integration.

Dill and colleagues recently reported empirical evidence that many Web applications share a regular structure.[6] Their main finding is that we can decompose the Web into cohesive collections of pages, tightly and robustly connected via a “navigational backbone,” with several other pages

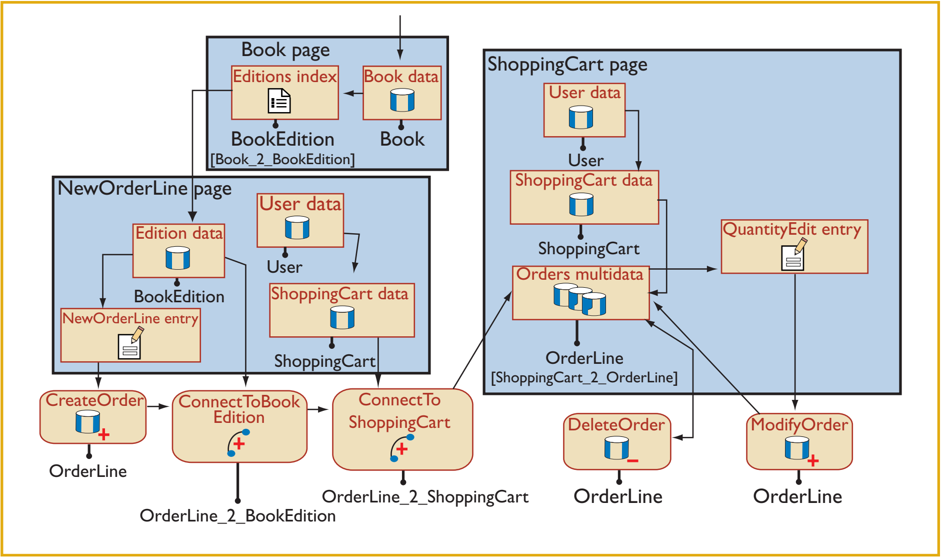


Figure 3. WebML content management operations. For the online book sales example, this hypertext fragment specifies the insertion of a new order into the user’s cart.

Table 2. WebML operation units.

Unit	Visual notation	Description
Create	Create unit +	Establishes a new instance of an entity.
Delete	Delete unit –	Removes an instance of an entity.
Modify	Modify unit ±	Changes an instance of an entity.
Connect	Connect unit +	Creates an instance of a relationship.
Disconnect	Disconnect unit –	Drops an instance of relationship.
Generic	Generic operation unit	Invokes a generic operation, possibly implemented by externally available Web services

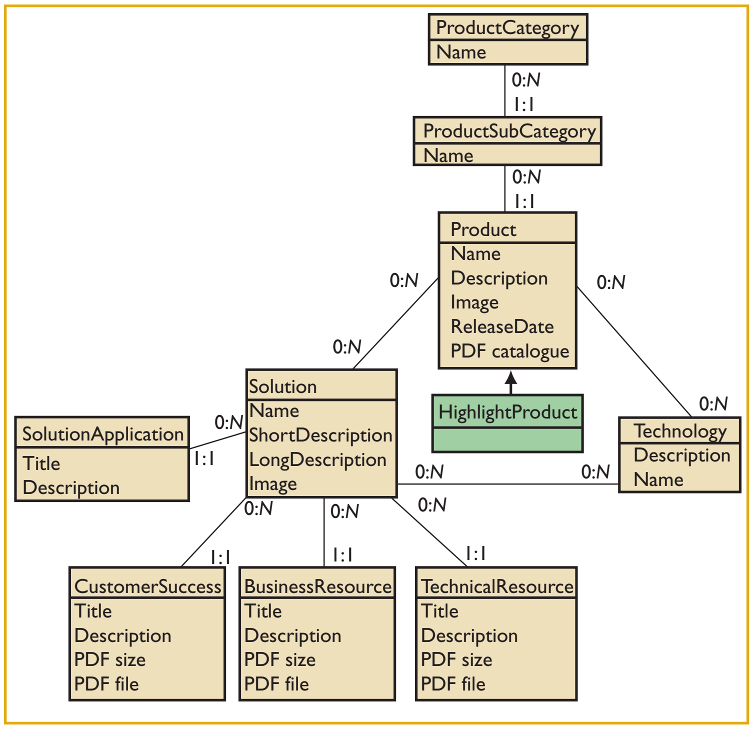


Figure 4. Simplified data schema for the Cisco Web site. Products, organized into categories and subcategories, are related to solutions and technologies; each solution connects to applications, success stories, and technical and business resources.

pointing *into* or reaching *out* of the backbone. This finding is surprisingly consistent with our framework: the backbone represents a connection schema including all core entities and the relationships among them; the *into* pages represent the access schema; and the *out* pages represent the core schema.

This structure is fractal in nature and thus repeats at various levels — the Web at large presents a structure similar to that of its cohesive portions. This suggests that "to design effective algorithms for data services at various scales of the Web, it is sufficient to understand the structure that emerges from one fairly simple stochastic process."[6] The strength of the Dill study is that the authors experimentally derived the reported properties of Web hypertext connectivity by analyzing a huge number of Web pages — about 60,000,000. However, they did not provide any semantic classification of the Web structures that emerged.

Figure 4 shows a simplified ER diagram of the Cisco institutional Web site (www.cisco.com), which we use to illustrate core, access, and interconnection skeletons. This site's goal is to publish information about the company's products, related technologies, and solutions. The ER diagram therefore includes three core entities that describe the core concepts: `Product`, `Solution`, and `Technology`; each has several attributes that show its properties. The diagram classifies products into categories and subcategories; the subclass `HighlightProduct` collects the best-selling products. Each solution has connections to several business and technical resources as well as to success stories and possible applications. Many-to-many relationships link core entities to one another.

Core Skeleton

A core skeleton helps the designer display and manage the site's most relevant business objects. Figure 5a shows a typical data subschema for the core skeleton, which normally has a hierarchical structure. The schema centers on one core entity, representing the core application object, and typically including attributes that represent its descriptive properties. In many cases, designers must substructure additional descriptive properties featuring multiple or structured values into internal components of the core entity. These can be, for example, multiple variants or multiple pictures of a given product or product documentation. Component entities must be connected to core entities (in relational terms, they include "foreign keys" of a referential integrity constraint).

The core schema's main function is to support content browsing — the user's wish to gather more information about the site's main application objects. Normally, users select core instances through the access schema and then navigate within the core schema to access its features. A typical hypertext for content browsing provides independent accesses from the core entity to the component entities. For example, the hypertext in Figure 5b uses WebML multidata and index units to show multiple instances of the two component entities. Also, users can browse instances of `ComponentEntity2` on a separate page through an indexed guided tour. They reach this tour through an index unit for accessing instances of `ComponentEntity2` and a scroller unit providing commands for scrolling between instances.

Figure 6 shows a page from the Cisco site that supports content browsing for the company's cable service provider solution and the WebML specification for that page. A data unit displays the solution's textual property. A multidata unit lists the multiple applications associated with the solution, and three links connect the solution page to other pages displaying customer successes, business resources, and technical resources. This core skeleton addresses the `Solution`, `SolutionApplication`, `CustomerSuccess`, `BusinessRe-`

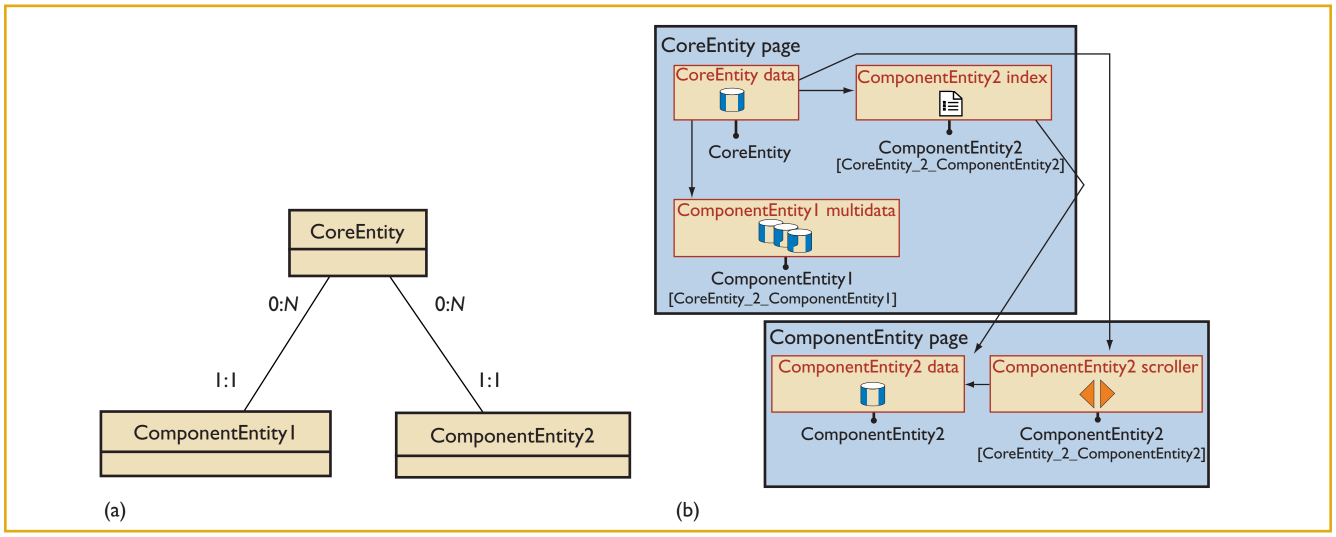
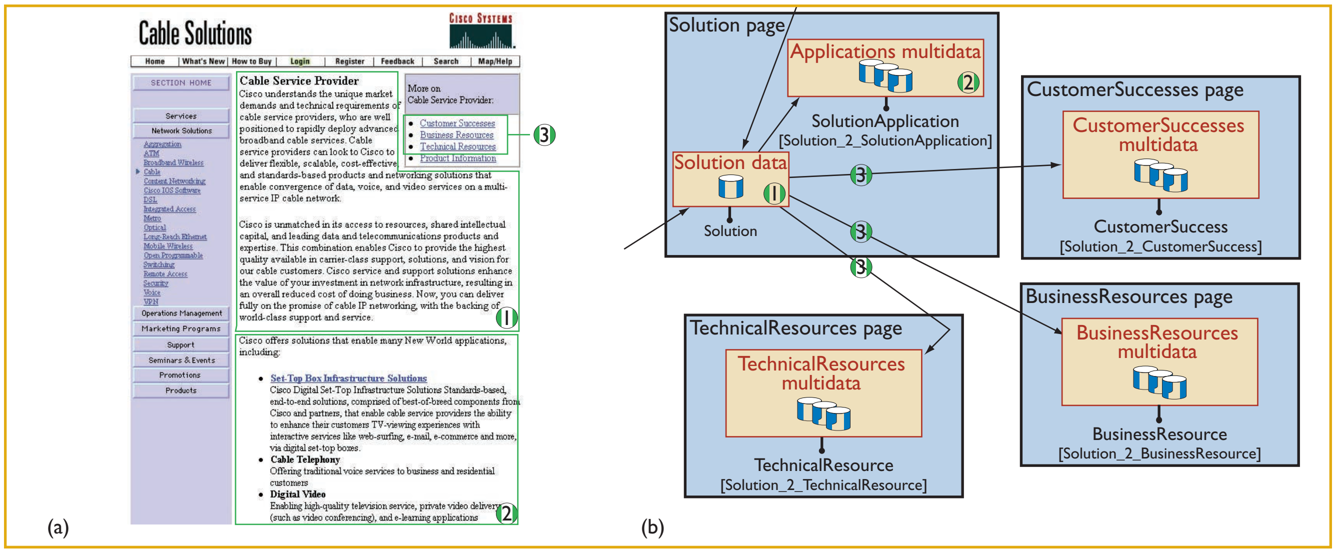


Figure 5. Core skeleton. (a) The data diagram represents the typical organization of a core concept into a core entity and several component entities. (b) The hypertext diagram includes a multidata unit for displaying ComponentEntity1 instances and an index and a scroller unit that realize an indexed guided tour for browsing ComponentEntity2 instances.


Cable Solutions

Cisco Systems

Home | What's New | How to Buy | Login | Register | Feedback | Search | Map/Help

SECTION HOME
Services
Network Solutions
Aggregation
ATM
Broadband Wireless
Cable
Content Networking
Cisco IOS Software
DSL
Integrated Access
Metro
Optical
Long-Reach Ethernet
Mobile Wireless
Open Programmable
Switching
Remote Access
Security
Voice
VPN
Operations Management
Marketing Programs
Support
Seminars & Events
Promotions
Products

Cable Service Provider

Cisco understands the unique market demands and technical requirements of cable service providers, who are well positioned to rapidly deploy advanced broadband cable services. Cable service providers can look to Cisco to deliver flexible, scalable, cost-effective and standards-based products and networking solutions that enable convergence of data, voice, and video services on a multi-service IP cable network.

Cisco is unmatched in its access to resources, shared intellectual capital, and leading data and telecommunications products and expertise. This combination enables Cisco to provide the highest quality available in carrier-class support, solutions, and vision for our cable customers. Cisco service and support solutions enhance the value of your investment in network infrastructure, resulting in an overall reduced cost of doing business. Now, you can deliver fully on the promise of cable IP networking, with the backing of world-class support and service.

More on Cable Service Provider:

- Customer Successes
- Business Resources
- Technical Resources
- Product Information

Cisco offers solutions that enable many New World applications, including:

- **Set-Top Box Infrastructure Solutions**
  Cisco Digital Set-Top Infrastructure Solutions Standards-based, end-to-end solutions, comprised of best-of-breed components from Cisco and partners, that enable cable service providers the ability to enhance their customers TV-viewing experiences with interactive services like web-surfing, e-mail, e-commerce and more, via digital set-top boxes.
- **Cable Telephony**
  Offering traditional voice services to business and residential customers
- **Digital Video**
  Enabling high-quality television service, private video delivery (such as video conferencing), and e-learning applications

Figure 6. Content browsing on the Cisco Web site. (a) This page from the Web site supports content browsing for the solution core concept. (b) The WebML specification shows a data unit displaying text about the solution, a multidata unit that shows associated applications, and links to other pages about technical resources, business resources, and customer successes.

`source`, and `TechnicalResource` entities from Figure 4.

Access Skeletons

We use access skeletons to select core concepts. Entities of an access skeleton typically represent categorization domains of the core concept. Categories must be treated as first-class concepts, not merely as property values of the core entity. This is because a category might itself store several pieces of information — a representative image and some descriptive text, for example, illustrating the common features of core concepts in the category.

Figure 7a (next page) shows a WebML data subschema for the access skeleton. The `CoreEntity`, which represents the core concept, is surrounded by two access entities `AccessEntity1` and `AccessEntity2`, which denote alternative categorization concepts. The diagram also contains `SpecialCollection`, which is a subentity of `CoreEntity` that denotes a collection of representative core concepts.

From the data standpoint, a typical access skeleton's organization recalls the classical schema of data marts, in which core data (facts) are the cen-

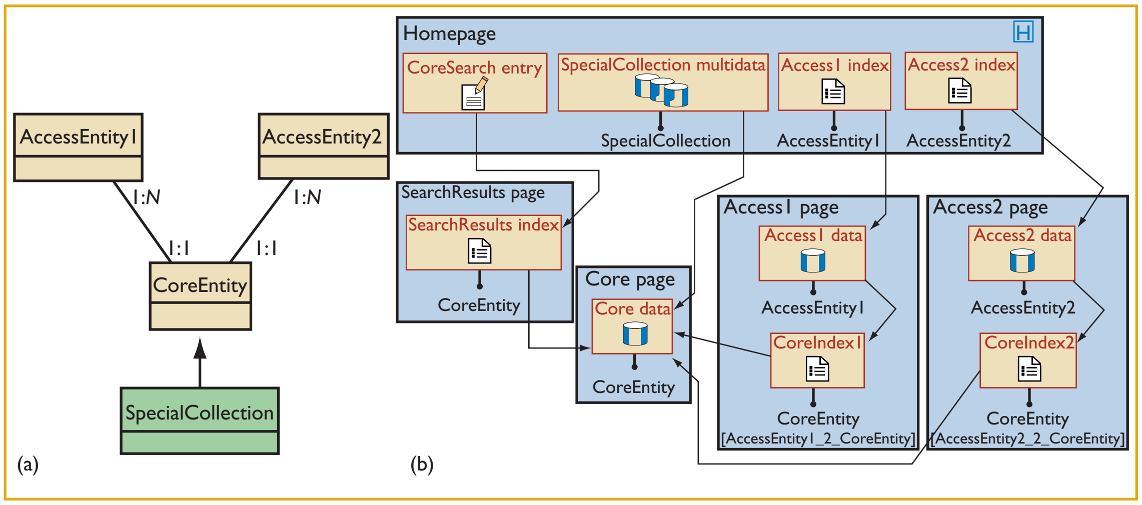


Figure 7. Access skeleton. (a) The data diagram includes access entities for categorizing core concepts and a subentity for denoting a special core concepts collection. (b) The hypertext diagram specifies different contextual and noncontextual access paths that depart from the Home page and lead to core data.

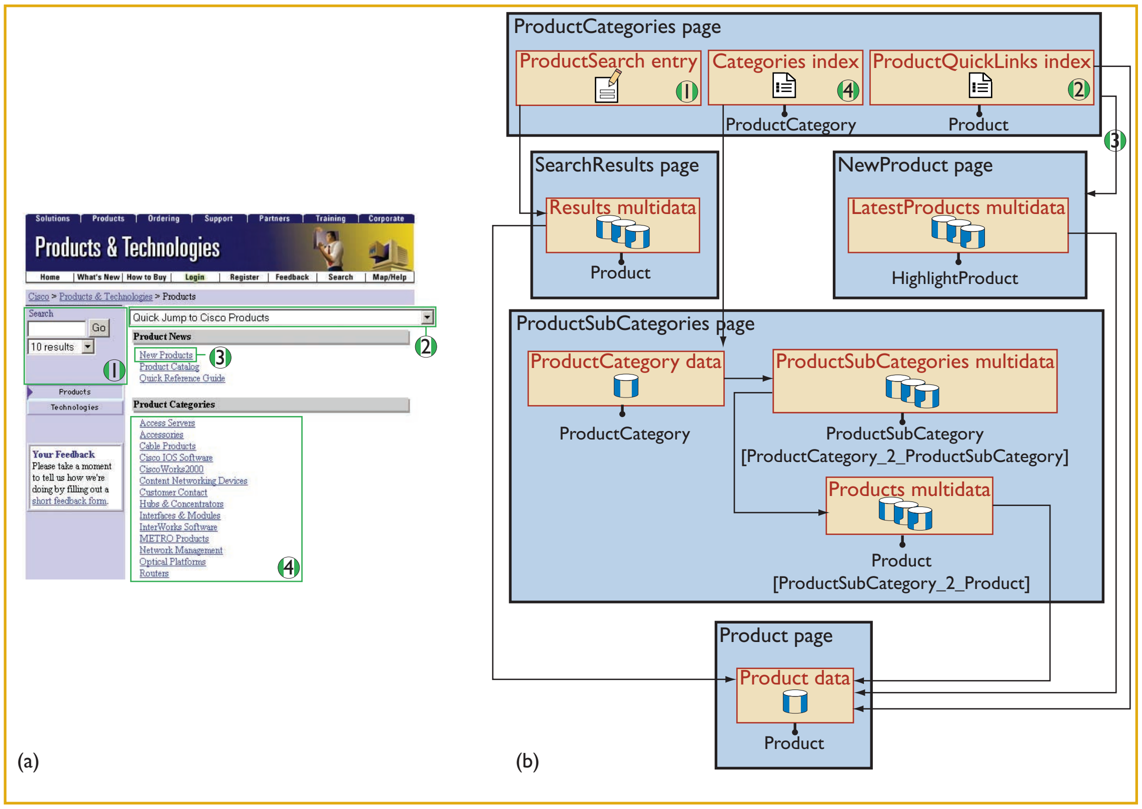


Figure 8. Product access on the Cisco site. (a) The Product Categories page supports access to individual products. (b) The WebML product access skeleton includes contextual and noncontextual access points.

ter of several "access dimensions."[7] Each access dimension can be arranged as a single entity or as a hierarchy of interrelated entities. The single-entity option results in a "star schema" structure; the hierarchy option results in a "snowflake schema." Similarly, within access skeletons, core entities lie at the center of several categorizing entities. The analogy extends not only to the schema's topology, but also to the use paradigm: in data warehouses we use software to effectively select facts according to several dimensions, thus performing data analysis. In the Web, we use categorizing entities to dynamically access core objects according to several alternative categorizations.

The hypertext expressing an access skeleton includes pages and content units dedicated to

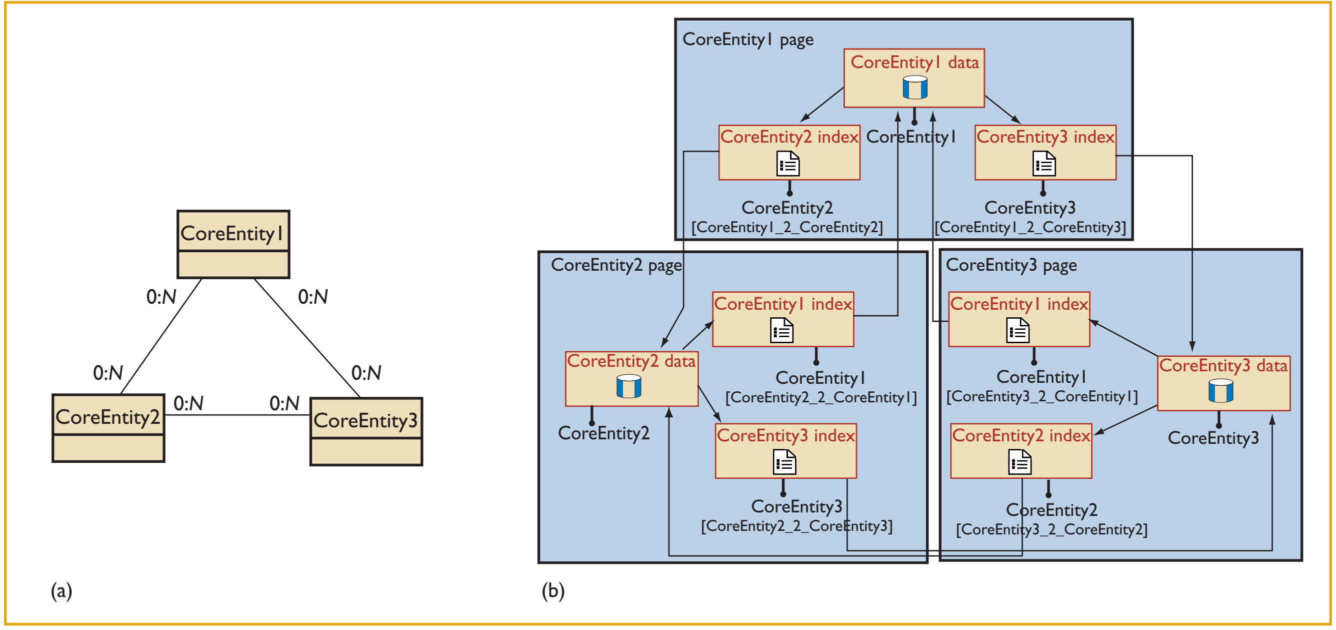


Figure 9. Interconnection skeleton. (a) The data diagram features many-to-many relationships defined between pairs of core entities. (b) The hypertext diagram depicts navigation paths for moving among core entities along the defined interconnections.

providing methods for locating core concepts. We categorize such access methods into two broad paradigms:

- *Contextual* access occurs when the user zooms in along a sequence of category-subcategory concepts, progressively moving from broad to narrower categories until locating the core concept of interest.
- *Noncontextual* access occurs when the user directly searches for the desired concepts by submitting keywords. This paradigm does not preserve a context between search and target pages.

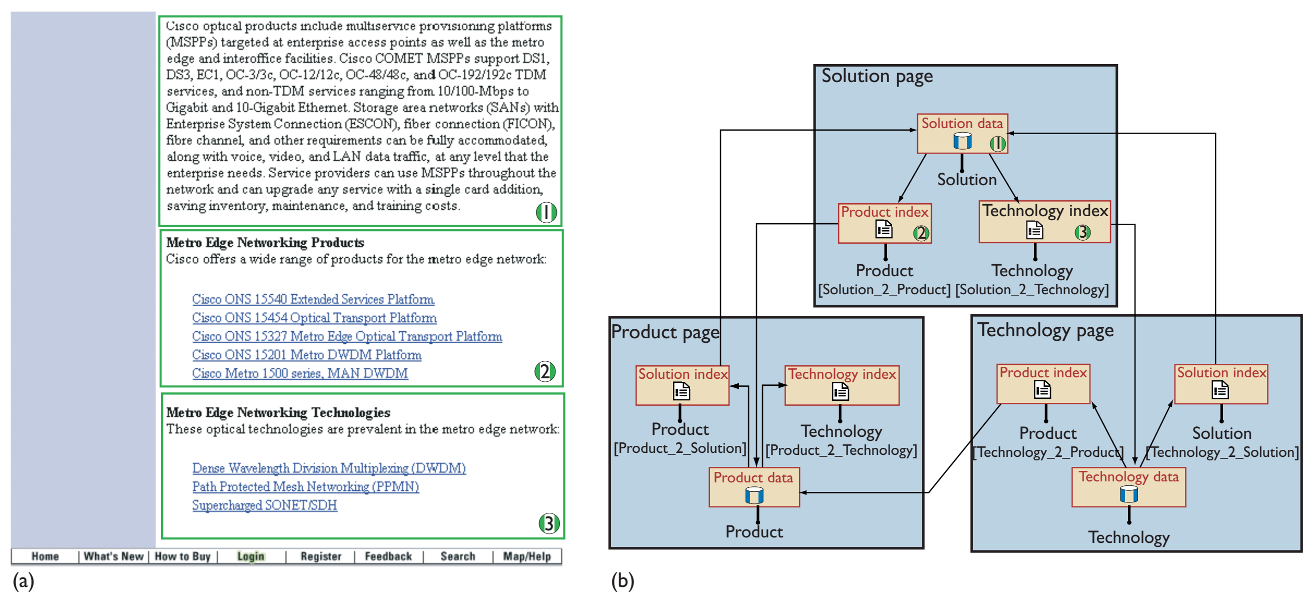
Figure 7b shows the typical topology of a hypertext fragment for an access skeleton that includes both contextual and noncontextual access methods. The `Homepage` includes two indexes over the categorization entities `Access1` and `Access2`, the starting points for two independent contextual navigation chains. The `Homepage` also contains an entry unit (`CoreSearch` entry) that denotes a search form for starting noncontextual access. Finally, the `Homepage` contains a multidata unit showing a collection of representative core concepts, from which users can jump to a page showing the full details of one of the concepts in the collection.

Figure 8 shows the Product access skeleton in the Cisco Web site, which addresses the `ProductCategory`, `ProductSubCategory`, `Product`, and `HighlightProduct` entities from Figure 4. Figure 8a corresponds to the `ProductCategories` page in the access skeleton shown in Figure 8b. The `ProductCategories` page includes an entry unit for a keyword-based search of Cisco products, a link to a compact index of all products, a link to new products, and an index of the product categories. Thus, the page supports one noncontextual access and provides starting points for three contextual accesses. The entry unit lets a user insert a search condition; the `SearchResults` page then displays a multidata unit that lists several products that meet this condition, and users select a single product from among them. The quick index points directly to products data, shown in the `Product` page. Users can also follow a link to the `NewProducts` page, on which a multidata unit shows the latest released products. Finally, the `Categories` index points to a page showing the `ProductCategory` data unit and a `ProductSubCategories` multidata unit, enabling two-step contextual access to products.

Interconnection Skeleton

Interconnection skeletons let us define relationships between a site's most relevant business objects. If each core concept is independent, the connection skeleton isn't relevant.

In the most general data skeleton, shown in Figure 9a each core entity relates to every other. A binary relationship exists between each pair of core entities, yielding a fully connected graph; unless otherwise specified, relationships are many-to-many. The corresponding hypertext, shown in Figure 9b, consists of three identical pages – one for each core entity. Each page includes a data unit



Cisco optical products include multiservice provisioning platforms (MSPPs) targeted at enterprise access points as well as the metro edge and interoffice facilities. Cisco COMET MSPPs support DS1, DS3, EC1, OC-3/3c, OC-12/12c, OC-48/48c, and OC-192/192c TDM services, and non-TDM services ranging from 10/100-Mbps to Gigabit and 10-Gigabit Ethernet. Storage area networks (SANs) with Enterprise System Connection (ESCON), fiber connection (FICON), fibre channel, and other requirements can be fully accommodated, along with voice, video, and LAN data traffic, at any level that the enterprise needs. Service providers can use MSPPs throughout the network and can upgrade any service with a single card addition, saving inventory, maintenance, and training costs. 1

Metro Edge Networking Products
Cisco offers a wide range of products for the metro edge network:

Cisco ONS 15540 Extended Services Platform
Cisco ONS 15454 Optical Transport Platform
Cisco ONS 15327 Metro Edge Optical Transport Platform
Cisco ONS 15201 Metro DWDM Platform
Cisco Metro 1500 series, MAN DWDM 2

Metro Edge Networking Technologies
These optical technologies are prevalent in the metro edge network:

Dense Wavelength Division Multiplexing (DWDM)
Path Protected Mesh Networking (PPMN)
Supercharged SONET/SDH 3

Figure 10. Core entities' interconnection within the Cisco Web site. (a) The Solution page shows data about the Metro Edge networking solution, and includes links to the connected products and technologies. (b) The WebML hypertext includes a page for each core entity; each page features a data unit defined on the core entity and two index units pointing to the associated instances of the other two core entities.

that displays one instance of the core entity (previously selected through the access skeleton) plus two index units for displaying related concepts from the other two core entities.

Figure 10 shows a Cisco Web site's `Solution` page along with the site's interconnection skeleton, addressing the `Product`, `Technology`, and `Solution` entities from Figure 4 and the relationships interconnecting them. The `Solution` page (Figure 10a) includes a data unit showing a description of a specific solution and two indexes pointing to the associated products and technologies.

Content Management Skeleton

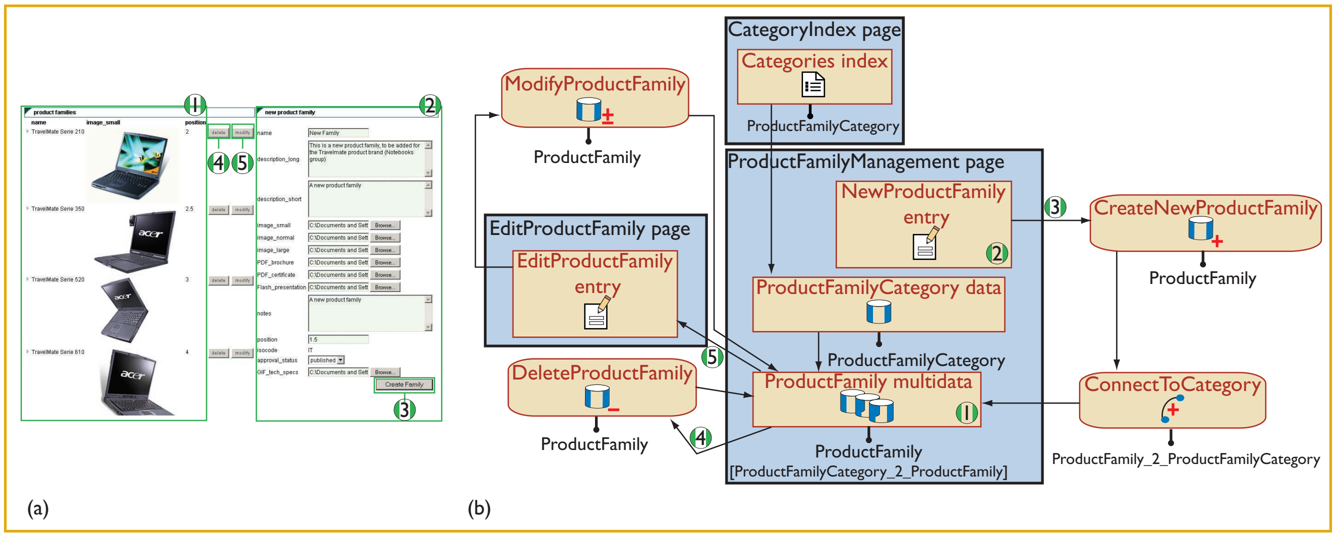
In addition to the previous data access skeletons, many Web applications support content management. This can either be restricted to administrators and content owners, who might use it to update product data in a corporate Web site, or exposed to end users, who might use it to insert messages in Web forums or manage their user profile data.

The content to be managed — which the Web application will publish dynamically — does not generally include the access schemas' entities, which are predefined, stable information for core content categorization. Content management concerns the insertion, deletion, or modification of core entities and related components (according to the core subschema) and the connection of such entities to access categories (according to the access subschema) and to other core entities (according to the interconnection subschema). Therefore, typical hypertexts for content management let us create one core and several component entity instances and then relate them to entity instances of the access and interconnection subschema. Such hypertexts are largely based on a typical WebML pattern of operations called the create-and-connect chain, which lets us generate a new entity instance and connect it to one or more existing instances.

To illustrate the content management skeleton, we'll look at the Acer Europe site (www.acer-euro.com). This Web application's goal is to serve both internal personnel and customers of the Acer European branch by organizing, collecting, managing, and publishing Web content about Acer products.

The page shown in Figure 11a is available to the application administrators for adding data about new product families to the Web site. The left side of the page shows a multidata unit including product families in the Travelmate category. The right part of the page contains a data entry unit for entering information about a new product family, with several fields available to administrators. The `Create Family` button activates a chain of create-and-connect operations that creates the new entity instance and connects it to the appropriate category. In addition, from each product family instance, buttons activate a delete unit for disposing of an existing family and a chain of data entry and modify units to change its content.

Figure 11b shows the WebML content manage-



product families

name	image_small	position		
TravelMate Serie 210		2	delete	modify
TravelMate Serie 350		2.5	delete	modify
TravelMate Serie 520		3	delete	modify
TravelMate Serie 610		4	delete	modify

new product family

name	New Family
description_long	This is a new product family, to be added for the Travelmate product brand (Notebooks group)
description_short	A new product family
image_small	C:\Documents and Sett Browse...
image_normal	C:\Documents and Sett Browse...
image_large	C:\Documents and Sett Browse...
PDF_brochure	C:\Documents and Sett Browse...
PDF_certificate	C:\Documents and Sett Browse...
Flash_presentation	C:\Documents and Sett Browse...
notes	A new product family
position	1.5
isocode	IT
approval_status	published
GIF_tech_specs	C:\Documents and Sett Browse...

Create Family

Figure 11. Content management on the Acer Web site. (a) The Product Family Management page supports the management of product families within a selected product family category. (b) The WebML specification includes a create-and-connect chain for generating a new product family in the selected category and delete and modify operations for removing or updating a product family in the category.

ment hypertext. Once the user selects a category of product families, the Web site uses a multidata unit to show its elements. Two buttons activate a delete unit or a chain of a data entry and a modify operation. The `NewProductFamily` entry unit includes a button for activating a create-and-connect chain, which generates a new product family and includes it in the previously selected category.

Using Skeletons

Skeletons provide an effective reasoning paradigm for understanding the deep structure of Web sites. A simple method that takes advantage of skeletons to design data-intensive applications might consist of the following steps:

- Determine the core concepts to be supported — the main application objects.
- Build the overall Web site data schema — access and core subschemas for each core concept and one interconnection subschema to relate core concepts if necessary.
- Build the required hypertexts — develop the access and core hypertext fragments for each core concept and link them through the connection hypertext fragment.

Skeletons also prove useful for application reengineering, which is necessary when an existing application becomes too inefficient or cumbersome to be managed. Reengineering is also the first step of Web integration, in which several separately designed applications must be accessed as one; in this case, it is very useful for the applications to follow the same logical organization.

The skeleton framework can also help us classify Web applications. We can regard Web sites as *access-centered* when they focus on supporting a rich variety of access dimensions, or as *core-centered* when they focus on supporting and managing core information. In addition to this initial classification, we can look at the Web site's core objects to identify the business model. We distinguish among five kinds of sites:

- *Commerce* sites sell the core object (B2B, B2C, e-shops, e-malls, virtual marketplaces, e-auctions).
- *Content* sites give users information about the core object (digital libraries, online magazines, recommending systems).
- *Service* sites offer the core object as a service (order-tracking sites and so on).
- *Community* sites build socially shared core objects (forums, chat rooms, newsletters, reselling systems).
- *Context* sites help to locate core objects (directories, search engines over local data).

These classes are almost orthogonal, and some sites combine them. Table 3 (next page) shows URLs of representative sites for some of the combinations.

We have employed a WebML-based modeling exercise with hundreds of graduate students in our laboratory to teach them to recognize the deep organization of actual Web applications. Readers interested in this experience can start by identifying the skeletons we described in the WebML Overview sec-

Table 3. Proposed taxonomy for data-intensive Web applications.

Business model	Access-centered	Core-centered
Commerce	CD Now	www.cdnow.com
	eBay	www.ebay.com
Content	Periódico El Mundo	www.elmundo.com
	Epinions	www.epinions.com
Service	DHL Worldwide Express	www.dhl.com
	Hotmail	www.hotmail.com
Community	Java developer	developer.java.sun.com/developer/community/community discussion
	Lycos Finance Stocks and News	www.quote.com
Context	Google	directory.google.com

tion for the Amazon site subset and then apply a similar exercise to the sites listed in Table 3.

Conclusions

Abstracting data-intensive sites in terms of generic skeletons has applications in Web design, reengineering, and classification. Various design tools can support the use of skeletons, but the conceptualization remains powerful regardless of the specific design environment in use.

We derived most of the basic concepts of skeletons from the concrete application of the WebML model and approach to several applications, including the design of an experimental version of the Cisco Web application, and the development of the commercial Acer Europe Web application. WebML is currently used within the WebRatio tool suite (www.webratio.com). Our current research activities on WebML include integrating WebML with workflow management and e-services and developing vertical application frameworks based on the skeletons presented in this article for a virtual campus project and business-to-business applications in e-logistics.

Acknowledgments

We thank the huge team of WebML designers and developers, particularly those who cooperated with us in application development. Our business model classification was inspired by a seminar given by A. Rangone at Politecnico di Milano in December 2000.

References

1. P. Fraternali, "Tools and Approaches for Developing Data-Intensive Web Applications: A Survey," *ACM Computing Survey,* vol. 31, no. 3, Sept. 1999, pp. 227-263.
2. S. Ceri, P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): A Modeling Language for Designing Web Sites," *Computer Networks,* vol. 33, nos. 1-6, June 2000, pp. 137-157.
3. G. Rossi et al., "Engineering Web Applications for Reuse," *IEEE Multimedia,* vol. 8, no. 1, Jan./Feb. 2001, pp. 20-31.
4. V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," *Proc. 27th Int'l Conf. Very Large Databases* (VLDB 01), Morgan Kaufmann, San Francisco, 2001, pp. 109-118.
5. S. Ceri et al., "Modeling Data Entry and Operations in WebML," *Proc. 3rd Int'l Workshop Web and Databases* (WebDB 00), Springer-Verlag, Berlin, 2000, pp. 201-214.
6. S. Dill et al., "Self-Similarity in the Web," *Proc. 27th Int'l Conf. Very Large Databases* (VLDB 01), Morgan Kaufmann, San Francisco, 2001, pp. 69-78.
7. R. Kimball, *The Data Warehouse Toolkit,* John Wiley & Sons, New York, 1996.

Stefano Ceri is a professor of database systems at Politecnico di Milano. His research focuses on data distribution, deductive and active rules, object-orientation, and design methods for data-intensive Web applications. He has been an associate editor of *ACM Transactions on Database Systems* and is currently an associate editor of several international journals. He was General Program Chair of VLDB 2001, and won the VLDB 2000 "Ten Years" award.

Piero Fraternali is an associate professor of software engineering at Politecnico di Milano. His research focuses on active rules, object orientation, design methods for data-intensive Web applications, CASE tools for automatic Web site production, and wireless applications. He is the main software architect of the WebRatio Tool Suite.

Maristella Matera is an assistant professor at Politecnico di Milano, where she teaches computer science fundamentals. Her research interests span design methods for Web and hypermedia applications, Web and hypermedia usability, and formal specification of interactive systems. She has been a visiting researcher at the Graphics, Visualization, and Usability Center at the Georgia Institute of Technology.

Readers can contact the authors at {ceri, fraterna, matera}@elet.polimi.it.