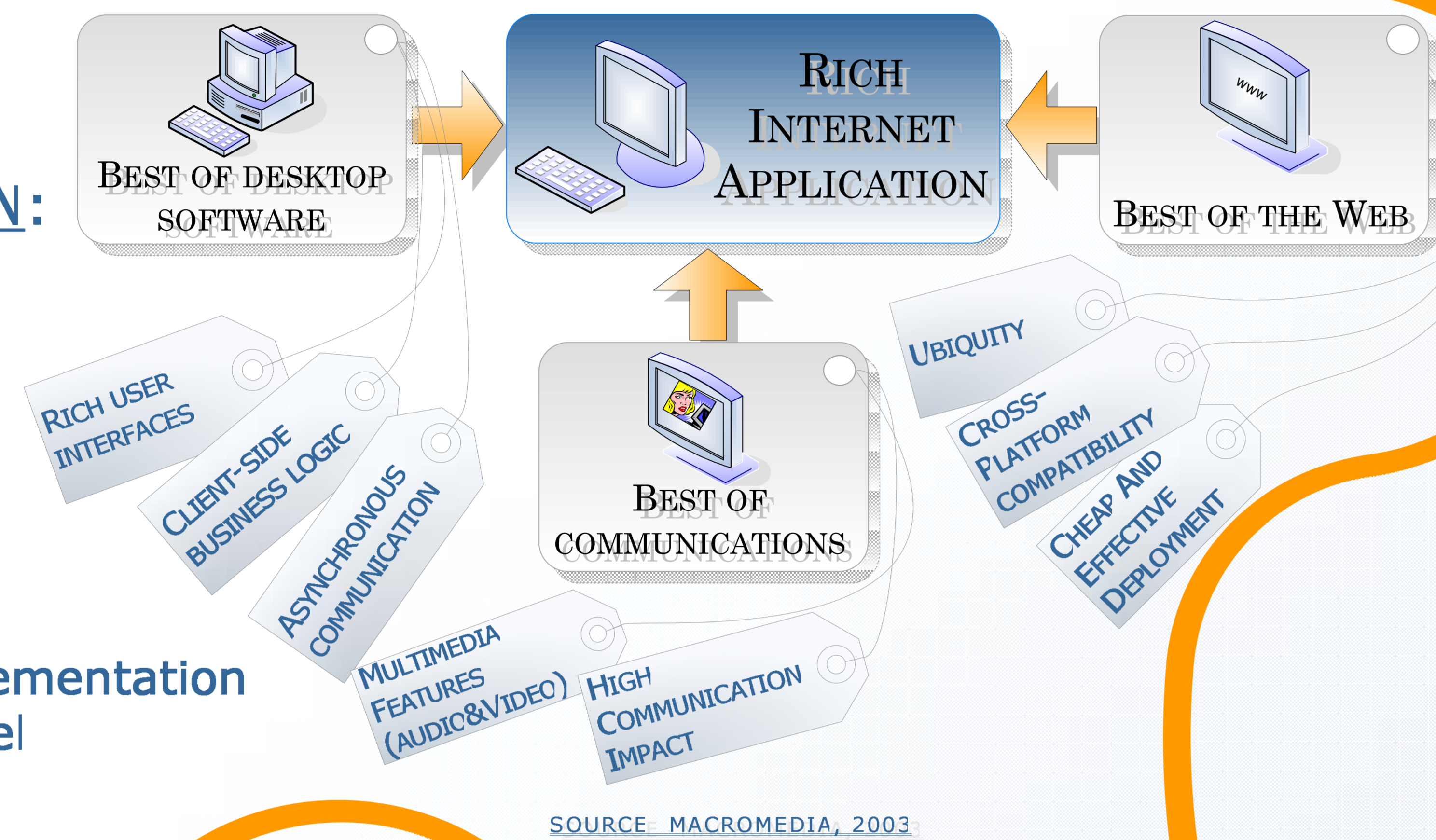# CAPTURING RIA CONCEPTS IN A WEB MODELING LANGUAGE

ALESSANDRO BOZZON :: SARA COMAI
PIERO FRATERNALI :: GIOVANNI TOFFETTI CARUGHI
POLITECNICO DI MILANO, DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE – MILANO, ITALY

1:: MOTIVATIONS

RICH INTERNET APPLICATIONS ARE A NEW BREED OF WEB APPLICATION...

➤ OFFERING SOPHISTICATED INTERFACES FOR REPRESENTING COMPLEX PROCESSES AND DATA
➤ IMPROVING PERFORMANCES BY MINIMIZING CLIENT-SERVER DATA TRANSFERS
➤ MOVING THE BUSINESS AND PRESENTATION LAYERS FROM THE SERVER TO THE CLIENT

OUR CONTRIBUTION:

➤ A revision of a Web modeling language (WebML) to make it applicable to RIAs
➤ A validation-by-implementation of the proposed model extensions.

OUR VISION:

➤ Leveraging a Web model to specify distributed applications

BEST OF DESKTOP SOFTWARE → RICH INTERNET APPLICATION ← BEST OF THE WEB

RICH USER INTERFACES / CLIENT-SIDE BUSINESS LOGIC / ASYNCHRONOUS COMMUNICATION / MULTIMEDIA FEATURES (AUDIO&VIDEO) / HIGH COMMUNICATION IMPACT / UBIQUITY / CROSS-PLATFORM COMPATIBILITY / CHEAP AND EFFECTIVE DEPLOYMENT

BEST OF COMMUNICATIONS

SOURCE: MACROMEDIA, 2003

2:: DATA MODEL EXTENSIONS

In RIAs content can reside in the client as main memory objects, or, with some RIA technologies, as persistent data.

RDBMS / XML-DB

DATA OF RIA APPLICATIONS ARE CHARACTERIZED BY THEIR ARCHITECTURAL TIER OF EXISTENCE AND BY THEIR LEVEL OF PERSISTENCE

S WISH ITEM — OID / CDTitle
S USER — OID / EMAIL / PASSWORD / USERNAME
S ORDER — OID / TOTAL
S CD — OID / ARTIST / LABEL / TRACK-NUMBER / PRICE / RELEASE DATE / TITLE
S TRACK — OID / TITLE / POSITION
S ORDER ITEM — OID
C WISH ITEM — OID / CDTitle
C CART — OID / TOTAL / ITEM-NUMBER
C CART ITEM — OID / CDTitle

- **S** Persistent Server Entity/Relationship
- **S** Volatile Server Entity/Relationship
- **C** Persistent Client Entity/Relationship
- **C** Volatile Client Entity/Relationship

DATA MODEL OF THE CD STORE CASE STUDY APPLICATION

3:: APPLICATION MODEL EXTENSIONS

Rich Internet Applications are characterized by:
➤ Having a different physical structure, consisting of a single application "shell" (e.g., a Java applet or a FLASH movie), which loads different data and components based on the user's interaction
➤ Splitting the computation between the server and the client, minimizing data transfers and reducing the server workload

THE HYPERTEXT MODELLING METAPHOR REMAINS A GOOD DESCRIPTION OF THE DYNAMICS OF THE INTERFACE ALSO FOR RIAs, ESPECIALLY IN THE CASE OF HYBRID APPLICATIONS

➤ The notion of page has been extended to cope with the specificity of RIAs

- **S** Server page/sub-page
- **C** Client page/sub-page

Data can be extracted and queried from both client and server entities/relationships

- **S** Server selector condition / order clause
- **C** Client selector condition / order clause

HYPERTEXT MODEL IN THE LARGE OF THE CD STORE CASE STUDY APPLICATION WITH AN EXAMPLE OF ITS RENDITION SOURCE LASZLOSYSTEMS.COM

HYPERTEXT MODEL IN THE SMALL OF THE CD STORE CASE STUDY APPLICATION WITH AN EXAMPLE OF ITS RENDITION SOURCE LASZLOSYSTEMS.COM

CLIENT — 4: Client-side data Query (Price selectors) → Client Component → 1: request CD Data / 3: Data response → Server Component → 2: Server-side data query (Title Selector) — SERVER
Data Tier

CLIENT-SERVER INTERACTION FOR THE HYPERTEXT MODEL IN THE SMALL OF THE CD STORE CASE STUDY APPLICATION

SPECIFICATION AND AUTOMATIC GENERATION OF COMPLEX WEB SOLUTIONS MIXING HTTP+HTML AND RICH INTERNET APPLICATIONS USING A SINGLE MODELING LANGUAGE AND TOOL

6:: FUTURE WORK & REFERENCES

Our work shows how to evolve present-day Web design methods and tools to support the requirements of Rich Internet Applications, and more generally of any Web-enabled technology that flexibly assigns functions and data to both the client and the server. This work is the initial step of a broader research program that will address several open issues:

➤ Enriching the model of the presentation layer
➤ Providing an integrated model for server-to-client interaction
➤ Revising the semantics of the conceptual model, to enable automated model-checking as presently done for conventional Web applications
➤ Distilling comprehensive guidelines supporting the design of RIAs
➤ Measuring performance and optimizing the generated code
➤ Providing an UML profile for dealing with RIAs

AUTHORS

➤ Alessandro Bozzon :: bozzon@elet.polimi.it
➤ Sara Comai :: comai@elet.polimi.it
➤ Piero Fraternali :: fraterna@elet.polimi.it
➤ Giovanni Toffetti Carughi :: toffetti@elet.polimi.it

Politecnico di Milano, Dipartimento di Elettronica e Informazione
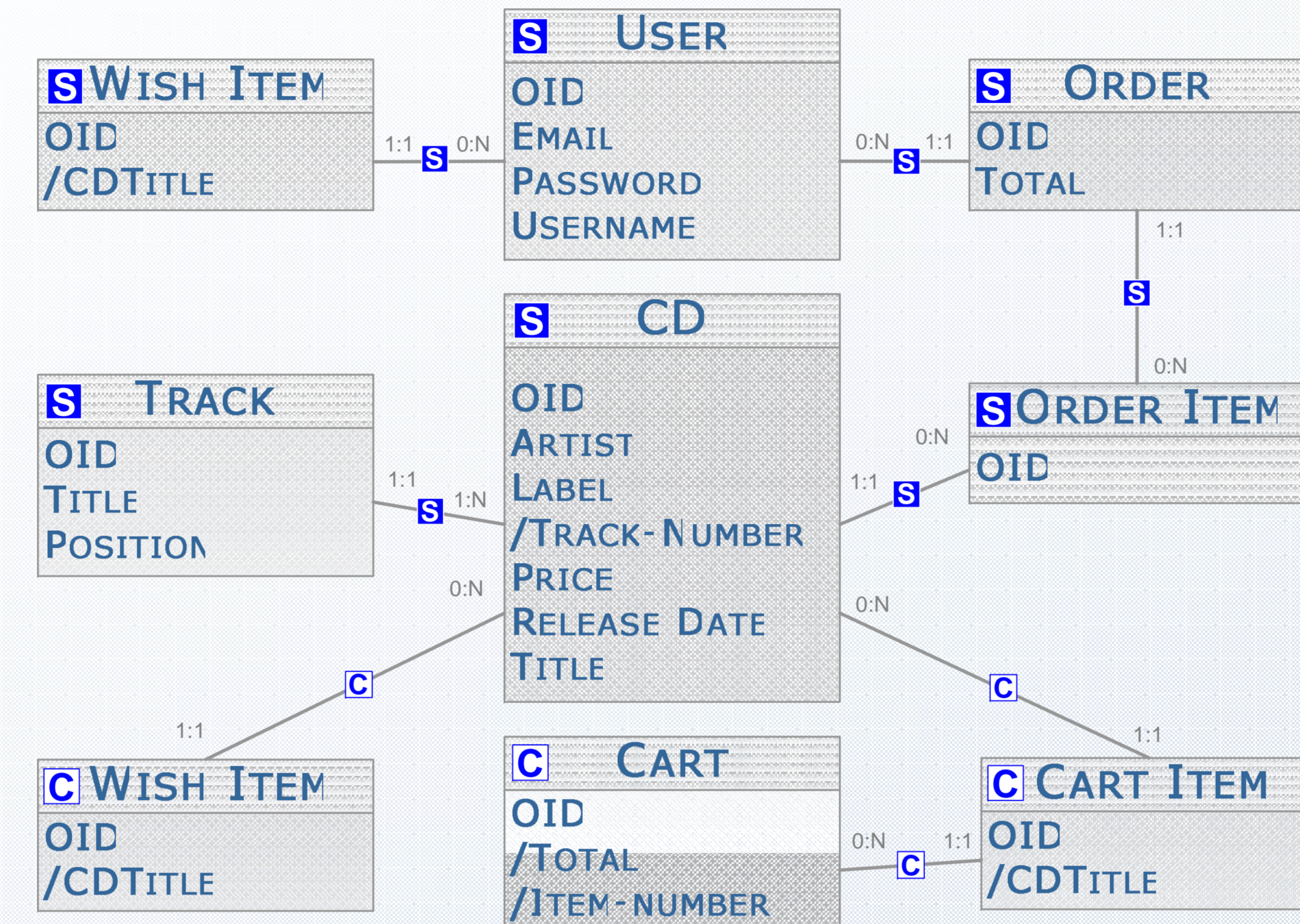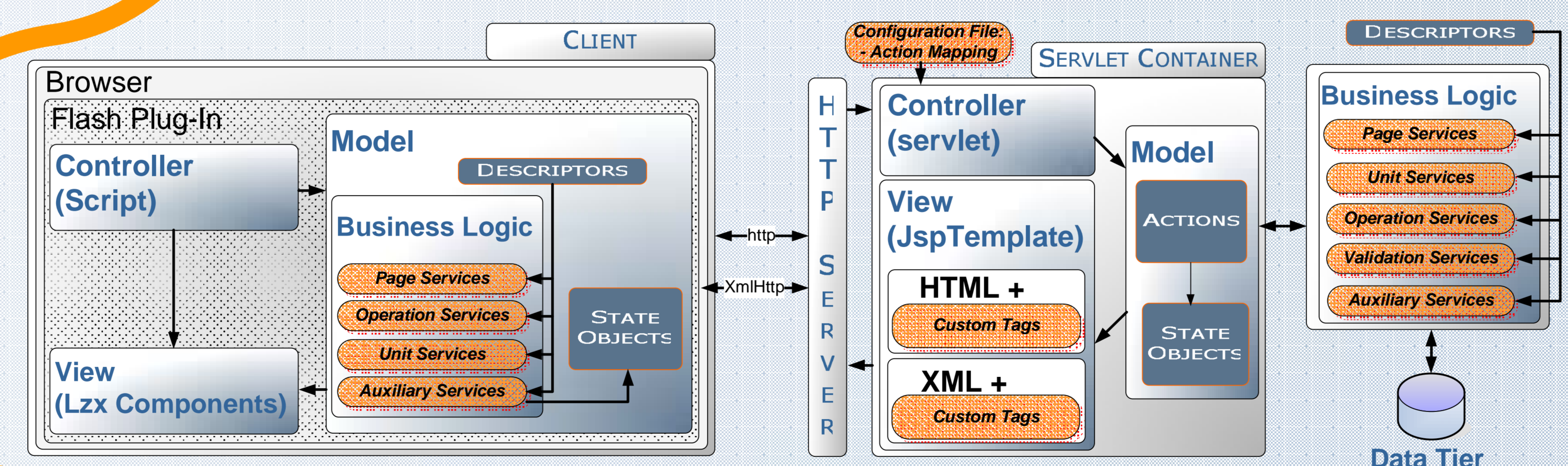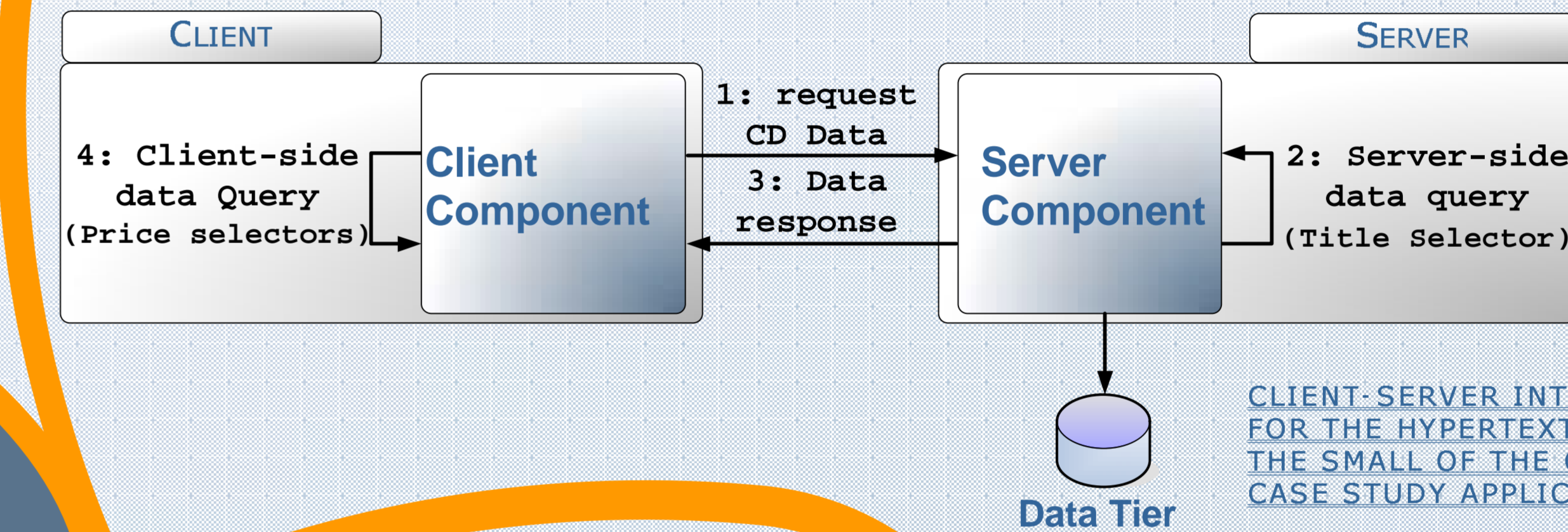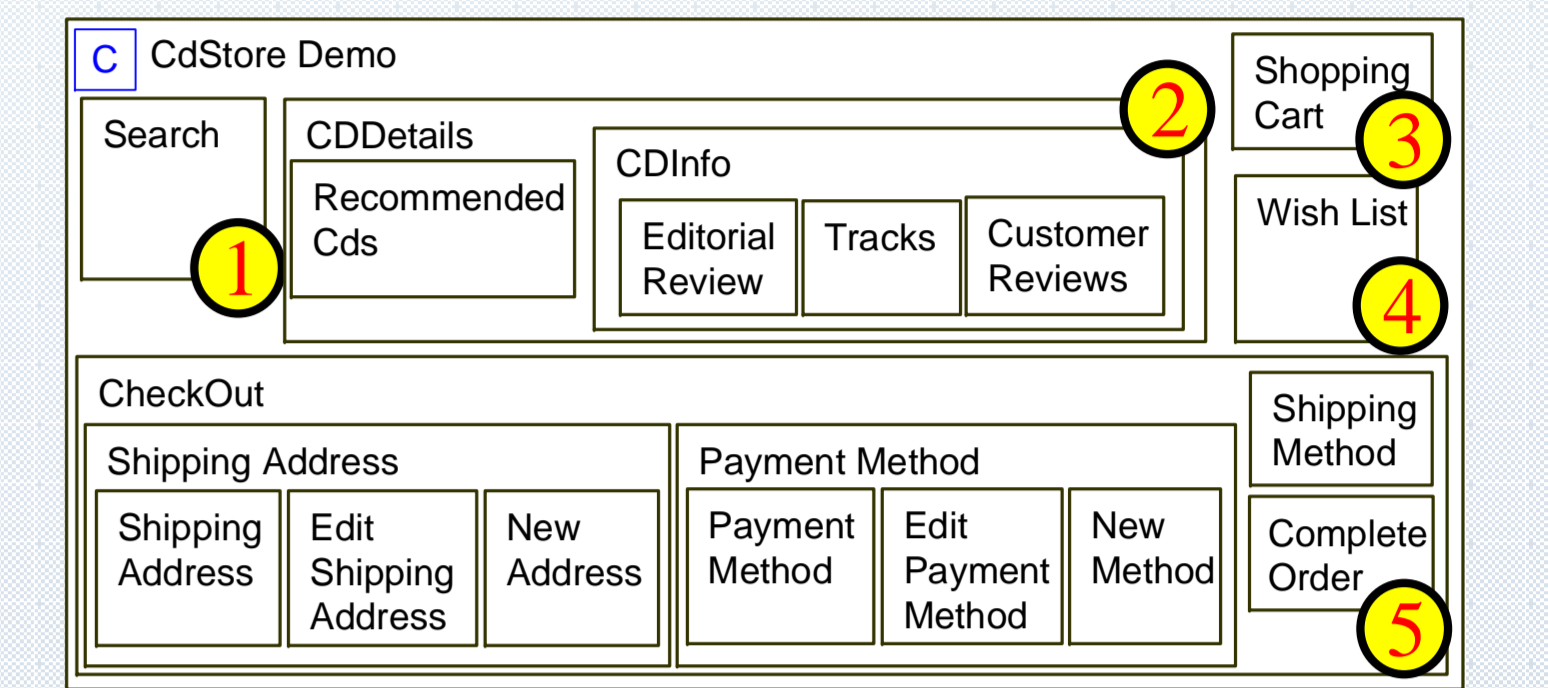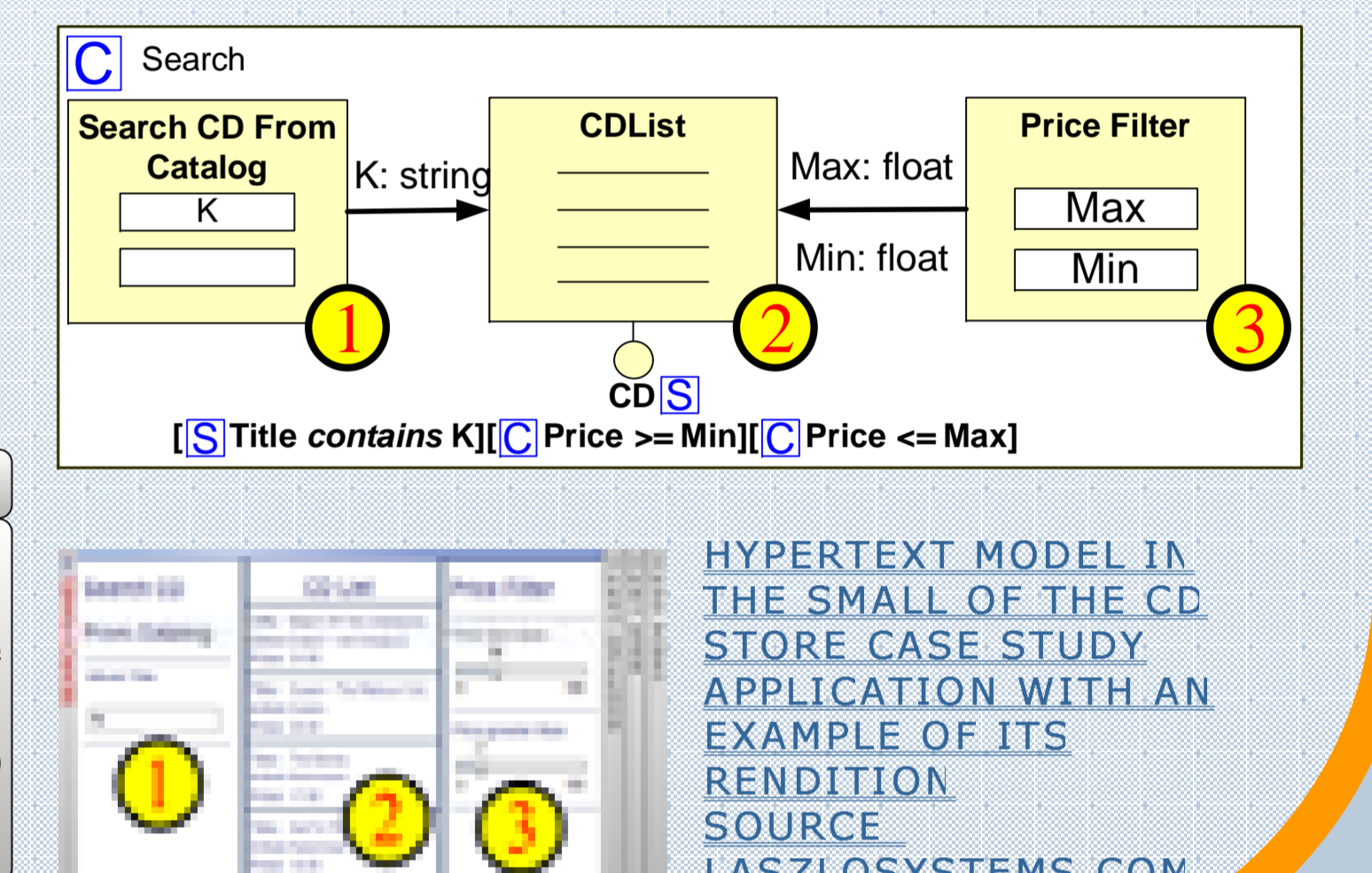P.zza L. da Vinci, 32 - Milano

REFERENCES

➤ WebML Website :: www.webml.org
➤ WebRatio Website :: www.webratio.com
➤ Designing Data-Intensive Web Applications
  :: S. Ceri, P. Fraternali, M. Brambilla, A. Bongio, S. Comai, M. Matera :: Morgan Kaufmann, 2002
➤ OpenLaszlo web site :: www.openlaszlo.org

5:: IMPLEMENTATION

IMPLEMENTATION EXPERIENCES HAVE BEEN PERFORMED WITH WEBRATIO, A CASE TOOL USING WEBML FOR THE VISUAL SPECIFICATION AND THE AUTOMATIC CODE GENERATION OF WEB APPLICATIONS.

CLIENT
Browser / Flash Plug-In
Controller (Script) / Model / DESCRIPTORS / Business Logic / Page Services / Operation Services / Unit Services / Auxiliary Services / STATE OBJECTS / View (Lzx Components)

H T T P S E R V E R — http / XmlHttp

SERVLET CONTAINER
Configuration File - Action Mapping
Controller (servlet) / Model / View (JspTemplate) / HTML + Custom Tags / XML + Custom Tags / ACTIONS / STATE OBJECTS

DESCRIPTORS
Business Logic / Page Services / Unit Services / Operation Services / Validation Services / Auxiliary Services
Data Tier

THE PROPOSED CLIENT-SERVER RUN-TIME ARCHITECTURE

Each WebML primitive is mapped onto run-time services, model objects and, where needed, view templates. WebRatio generates a single generic service for each type of element (unit, operation or page), both client and server. This generic service is configured by means of XML descriptor files, used at runtime to instantiate the (unique) generic service into as many concrete ones as necessary.

Adopting the same architecture for generating both the server-side and the client-side code it is possible to specify and automatically generate complex Web solutions using a single modeling language and tool.

4:: OPERATION MODEL EXTENSIONS

IN THE RIA CONTEXT, ARBITRARY BUSINESS LOGICS AND CONTENT UPDATE OPERATIONS CAN BE EXECUTED BY THE CLIENT, POSSIBLY WITH THE COLLABORATION OF THE SERVER.

➤ Server operation: a piece of business logic or data update executed by the server
➤ Client operation: a piece of business logic executed by the client (possibly with the participation of the server) or an update on a client-side entity or relationship
➤ Operation chain: a sequence of operations, possibly mixing client and server operations

- **S** Server operation
- **C** Client operation
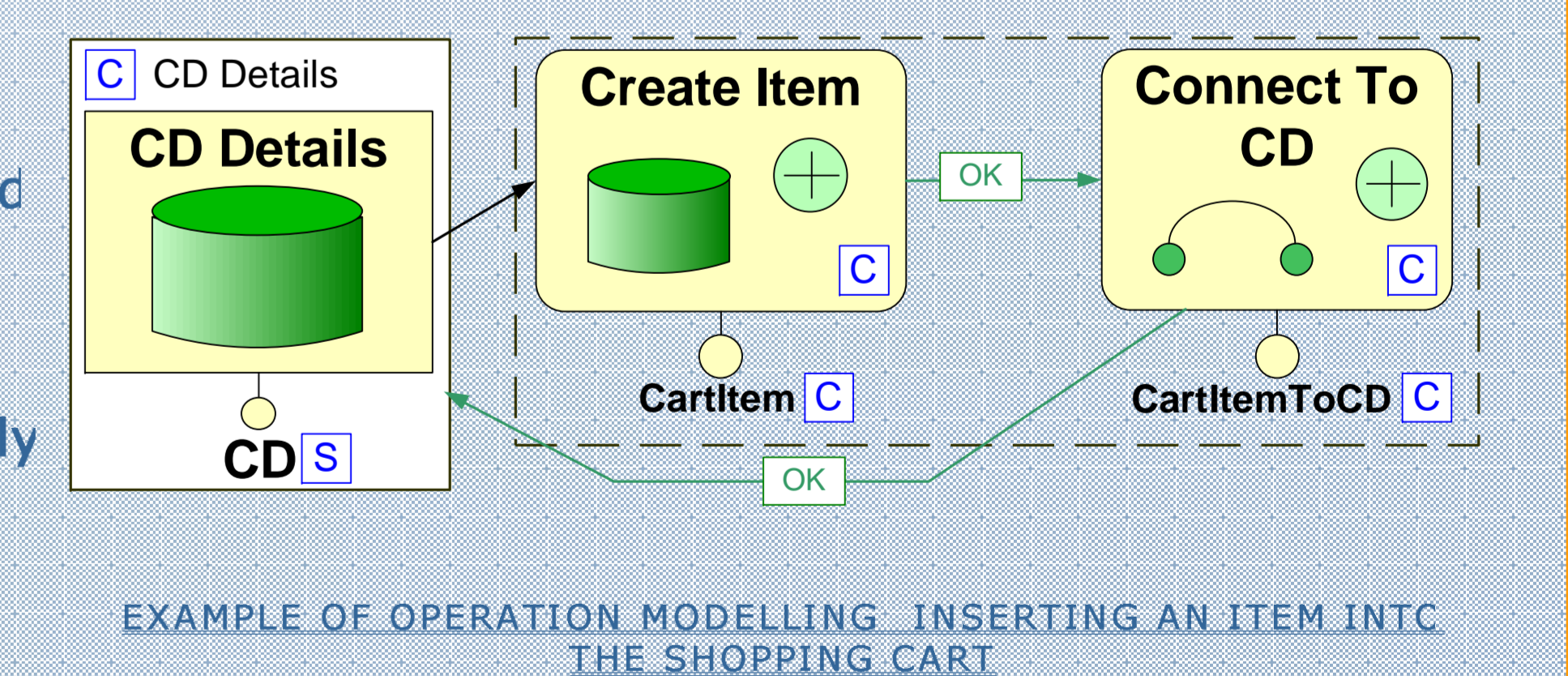
EXAMPLE OF OPERATION MODELLING INSERTING AN ITEM INTO THE SHOPPING CART

Data distribution between the server and the client arises the need of efficiently modeling bulk data transfers.

Transactional properties are not applicable to operation chains involving client and server operations. However, it remains possible to define transactional server or client blocks inside an operation chain.
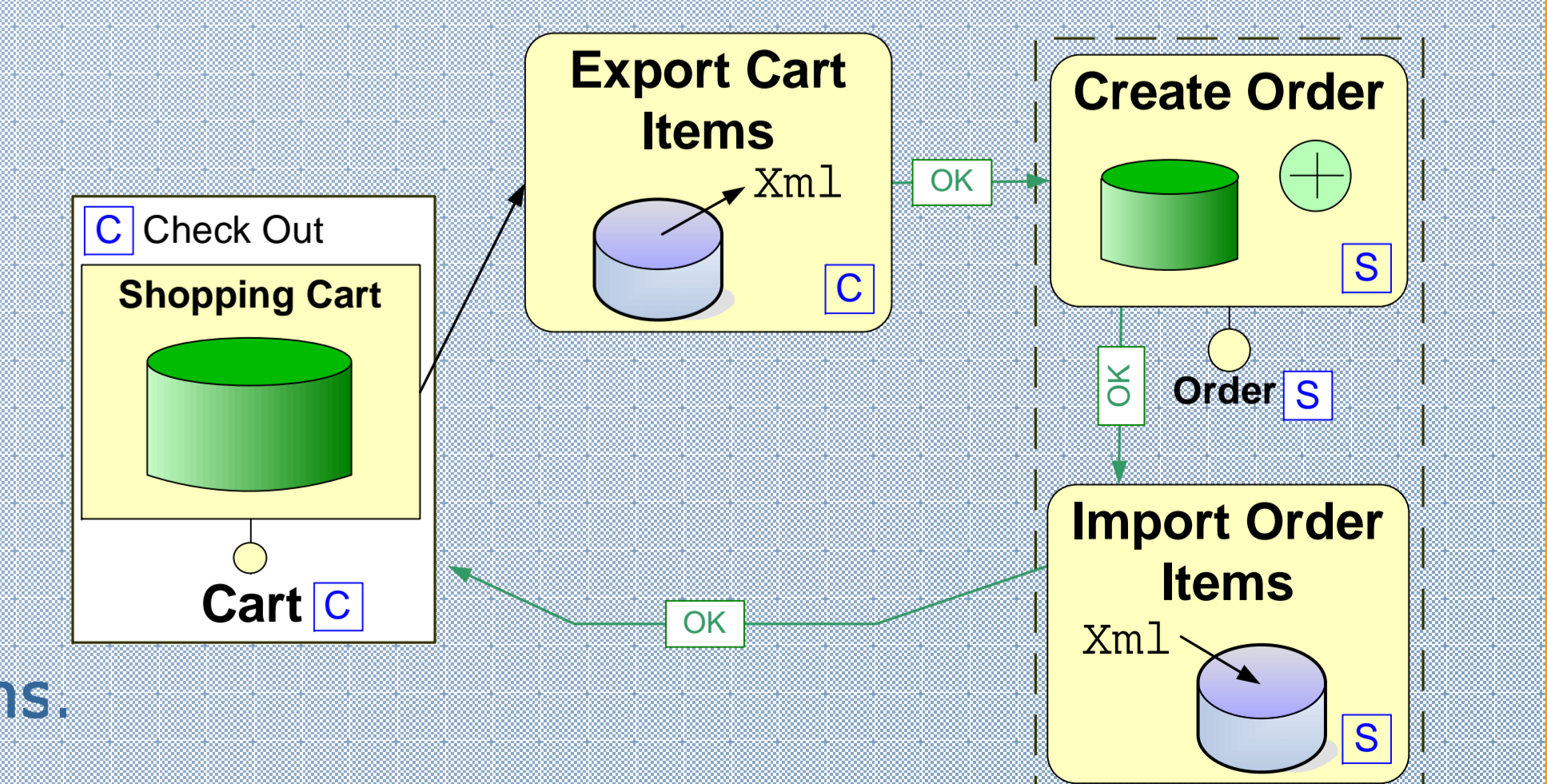
EXAMPLE OF OPERATION CHAIN FOR BULK DATA TRANSFER PLACING AN ORDER FROM THE SHOPPING CART