

# A Methodology for the Design of Distributed Web Systems

Guido Poncia and Barbara Pernici

Dipartimento di Elettronica e Informazione, Politecnico di Milano,  
Piazza Leonardo Da Vinci 32, 20133 Milano, Italy

**Abstract.** The increasing complexity of information systems and their interconnection to WWW applications make the design of such systems a difficult task. To produce large size web environments, systematic design methodologies and conceptual and design models have been developed. We present a methodology for the design of web systems in which servers are distributed over a wide geographical area. This context obliges us to consider not only the organisation of the information, but also its allocation to servers. The starting point is RMM (Relationship Management Methodology), developed for hypermedia environments. Since we consider a distributed environment, this methodology has been extended developing adequate tools and strategies. Particular attention has been devoted to the problem of data allocation. The allocation strategy is based on the fragmentation of the information and on the study of access dynamics. An example, based on an Intranet system, is presented.

## 1 Introduction

In these last months we are observing the progressive and rapid diffusion of Internet services, and in particular of the World Wide Web (W3) [3].

The fact that a constantly increasing number of people access everyday W3 sites from all over the world imply in some cases an excessive workload of the internet, with growth of the time spent for page retrieval. Technology improvement of hardware and software of communication systems surely can shorten such waiting times, but this kind of improvement is not immediate and its benefits are temporary. A solution to the problem can come from a systematic application of data distribution strategies, together with a design of the website based on a rigorous methodology.

One of the first attempt to improve the connectivity speed, the *mirroring*, consists in a replication of the whole page set on remote servers, the *mirrors*, located in different geographical areas. The fact that the information is entirely duplicated is a disadvantage, when some information is specific for a site. In this case, the full duplication is excessively redundant. Moreover, if the information is volatile, i.e., the server needs frequent updating, and all the mirrors have to be updated simultaneously in order to avoid data inconsistencies. In addition, a web system based on mirroring is not really distributed. Actually, the information is managed only on the main server, and loaded on remote servers only in a second phase.

Web systems of big corporations [11] or international organisations are structured on hundreds of servers, accessed worldwide, which contain thousands of pages. We can name these complex structures as *Distributed WebSite Systems* (DWSS).

In many organisations, the need of providing a complex set of information distributed over a large number of servers is not supported by a systematic design approach. Often, the single page design is extremely accurate, but the site has not a well defined structure. The approach commonly followed is a bottom-up approach, where the single page is the focus of the work, and the information to be provided is linked to others based on case by case decisions. In sites designed with a bottom-up approach a user has the risk of not finding the information he needs. This is the well known problem of *getting lost on the web*. Moreover, in a poorly designed DWSS, one can wait a remote downloading for a long time.

In recent times, the growth in the complexity of the information to be provided has imposed the need of a different approach. The systematic design of a web system based on a top-down methodology is now a subject of remarkable interest, and some proposals have emerged [1, 4, 9]. This kind of approach has been successfully employed in the field of the hypermedia design, with the development of adequate methodologies. We can mention here HDM [9] and RMM [1]. The first approach is based, among others, on the concept of *authoring-in-the-large*, i.e. the specification of global and structural aspects of the hypertext application. In HDM, links belong to different categories, and there is a distinction between *hyperbase*, that is the database of hypermedia, and the *access structure*. The definition of a HDM schema allows the integration of data and link structure. The final design of the page content is denoted as *authoring-in-the-small*. RMM is an evolution for hypermedia design of the entity-relationship approach. The information is structured in entities, and, in addition, relationships have the meaning of *access primitives*. We can anticipate here that this methodology has been chosen in this work as a basis for the design methodology for distributed systems.

In the field of DWSS, the important problem of distributing the information to different servers in order to organize satisfactorily the Web System and minimize the access time has never been solved through the elaboration of a systematic approach.

The aim of this paper is to present a methodology for the top-down design of a complex web system with distributed servers, defining in particular the criteria for the adoption of a distributed architecture, the main principles and rules of the distribution process. The original aspect of this methodology consists in an evaluation of access dynamics of users and consequently the partition of the information into fragments. In a second phase, the allocation process allows the designer to keep a decision for distributing fragments on local servers. Particular attention is devoted to the possibility of a partial data replication (redundant allocation) strategy, successfully adopted in the field of database design, improving the mirroring technique (see [5, 6] for details on database distribution methodologies).

After a description of features of DWSS architectures, with particular accent on the Intranets (Section 2), we will present the systematic methodology for the design of distributed Web Systems, focusing on phases that concern the problem of information distribution (Section 3). As preliminar application of the methodology, we will present a simple case study, namely the modelling of the Intranet of the sales department of a big company.

## 2 Distributed Architectures

Which are the guidelines that can allow us to move towards a distributed architecture of a web system? When is a top-down approach useful? The first step is to explore the main characteristics of the information system to be designed, in particular the aim of the application, the need of a frequent updating and refreshing (volatility), the structure of users and especially their distribution, the availability of adequate technologies.

This preliminary information can help us to decide for a distributed architecture or for a simpler structure based on mirrors or centralized sites. Obviously, when the information is created and retrieved from different sites distributed on a wide area (e.g. world-wide) we can be inclined towards a more complex distribution of information. Nevertheless, if the data provided do not need a frequent refreshment, a mirror system can be sufficient. Applications like DWSS are highly structured and have high information volatility: a systematic approach seems to be particularly appropriate. On the opposite side, the methodology is not useful for a simple WebSite, composed by few pages and in which information remains unchanged.

Providing a large amount of information produced world-wide to a well defined set of users geographically distributed is one of the needs of a big corporation. An increasing number of companies are now turning to Web as a way to organize their internal communication system [8], that is they have implemented or they are planning to implement Intranet systems. An Intranet can be defined as an internal web of a corporation accessed only from a well defined set of users, the corporation members. Generally, it is composed by a complex set of geographically distributed servers providing highly volatile information.

### 2.1 Intranets

An Intranet system is a way for a corporation to improve the internal communication system [7].

The field of application of Intranet systems is very broad [2], satisfying requirements in a corporation environment in areas such as Sales and Marketing, Product Development, Customer Services, Training and Human Resources applications. In addition, Intranets can be linked to the internet world and the possibility of retrieving external information is extremely large [12]. It is important to outline the fact that an Intranet system can lead to a dramatic reduction of costs of content development, duplication, distribution and usage. The full

usage of Intranets in a corporation is a step towards the introduction of the so called *paperless office*. Intranet technology can fully support a distributed computing strategy, allowing to place the servers and the content closer to the owner of that content.

The methodology presented in the following tries to give a systematic pattern that can be followed when an Intranet system is designed. The original aspect of this approach is that it copes with the problem of distributing information in order to improve the performance of the web system, with respect to the access time and update workload.

## 2.2 The Case Study

In the following, we will apply the methodology to a simple case of Intranet of a sales department of a corporation. We consider a company which has the headquarters in Europe and two big sales divisions in America and in Australasia. The Intranet system has to provide salesmen with an on-line and up to date information about:

- news of the day and warnings of the corporation headquarters;
- information about products, their characteristics, their technical description;
- the price lists of products, the conditions of sale;
- last news about products, particular offers, etc.;
- a description of customers;
- the dealers' homepages and their activity;
- training service and remote tutorial for young dealers.

Moreover, the Intranet system can give the possibility of a feedback to salesmen, and the automatic submission of a sale action.

This case study has the main advantage to take into consideration the main properties of an Intranet system in a simple way, even if the number of websites is small, and the information content is well defined.

## 3 A Systematic Approach to Distributed WWW Design

In a top-down approach for web design, first we have to understand from the user the requirements that the system has to satisfy. In the following, the development of a general model that describes the so called *in-the-large* logical configuration, and the *in-the-small* detailed structure is needed. The steps followed by the designer are the classical ones for the development of systems in the Software Engineering field [10, 15].

The methodology developed first considers the global organisation of the information (structure, user interaction) and then how this information can be distributed. Our approach wants to gather design methods typical of hypermedia design and W3 with the ones that consider data distribution issues in databases [5, 6]. The design process is conceptually branched in two sub-processes, the *Global Design* and the *Distribution Design*. These two different patterns are

anyway strictly related and they have to be followed in parallel and interactively. In fact, distribution criteria have to be generated taking into account global features of the site. The phases of the methodology are briefly presented in the following:

- (i) *Requirement analysis*: through a continuous interaction with users and providers of the information, we specify all the system elements, their localisation, and the requirements that have to be satisfied. We can distinguish between functional requirements, i.e., the background and objectives of users, and non-functional requirements, that is HW and SW and connection speed of the net [4].
- (ii) *Conceptual Design*: in this phase global schemata of data (e.g. E-R) and operations of users and providers (action model) are developed.
- (iii) *Global Logical Design*: a global schema that integrates data and access structure is designed on RMDM (Relationship Management Data Model) theory [13].
- (iv) *Distribution Specification*: the geographical allocation of servers and the classification of data in relation to volatility and locality of updates and retrievals are considered. Fragmentation and access tables are generated in order to collect the requirements for the distribution process.
- (v) *Distribution Design*: with the support of a decision tree, data allocation to servers is developed. The information can be distributed or not, and the possibility of data replication is evaluated.
- (vi) *Local Design*: the structure of each server is obtained merging the information of the RMDM global schema together with that derived from the distribution table. A schema of the modelled sites is produced.
- (vii) *Physical design*: in-the-small design of the single page is realized, with particular attention to the layout, the micro information content, the programming of HTML source and other applications ad scripts.

Assuming that the global design methods and models are procedures well described in literature, in particular E-R analysis, RMDM scheme derivation and operation schemas construction, in the following we will focus particularly parts of the methodology related to DWSS design.

### 3.1 Requirement Analysis and Conceptual Design

The first step of the design is understanding which are the system aims, that is what kind of information has to be put on-line and the users that will produce or use this information. The advantage of an Intranet is that we can suppose the set of users well defined, thus we can predict their behaviour and at the same time decide the services to provide.

A classification of the subjects involved is anyway necessary [4]. Criteria of segmentation are based not only on objectives and needs of users. In a distributed context the geographical and functional location are fundamental to decide which local views of the available information should be proposed. In the example, users

are subdivided according to their location, in Europe, America or Australasia sales area.

An additional segmentation of users, depending from their modalities of access or their function into the company environment is of course possible. We can, for instance, classify users by connection type, e.g. via modem or direct TCP-IP access. We will see in Section 3.3 how the type of connection will be taken into account for the evaluation of information access costs. The feasibility of a distributed system and the distribution politics depends hugely from the hardware structure, and in particular from the access speed of net links between nodes. A localisation of data appears to be necessary as the speed of connections decreases.

The organisation of the information provided has to be represented in a clear and understandable form, for example using the E-R model. In the E-R schema we specify the relationships that logically link different entities in order to acquire the general structure of the information.

### 3.2 Global Logical Design

For web systems it is also important to decide how these relations are realized in terms of modalities of information access. In RMM methodology [13] for hypermedia design entities are linked through types relations.

The RMDM model is used to design a global logical schema of the Web System in this phase, starting from the E-R schema developed in the previous phase (for a complete description of the model design we refer to [13]).

RMDM provides a language for describing the information objects and the navigation mechanisms in hypermedia applications. Information is modeled with *domain primitives*, that is *entities*, representing abstract or physical objects and *associative relationships*, describing associations among entities. We consider relationships characterized by *access primitives*, indicating the navigation modality. In RMM, *menu relationships* and *sequence relationships* are defined for access primitives. The first ones allow the user to choose among components of an entity simultaneously, while the last ones produce a link towards an entity and a sequential access from the first to the last component.

Entities are represented by rectangular boxes, while relationships are dotted arrows. The symbols used for access primitives are reported on the bottom of Fig. 1. A new symbol is introduced to denote possible external links connecting the site to the whole Internet.

The sales RMDM schema is reported in Fig. 1.

From the homepage the user can access through a menu the training programmes, the company competitors, the customers, the salesmen, the product information and the news lists. Sequentially, he can access the news of the day. From competitors one can access the related external websites. From a product page, product news and prices can be accessed in sequential mode. It is also possible to access the sales actions concerning that product. Similarly, one can access sales actions concerning a customer and regarding a salesman from the

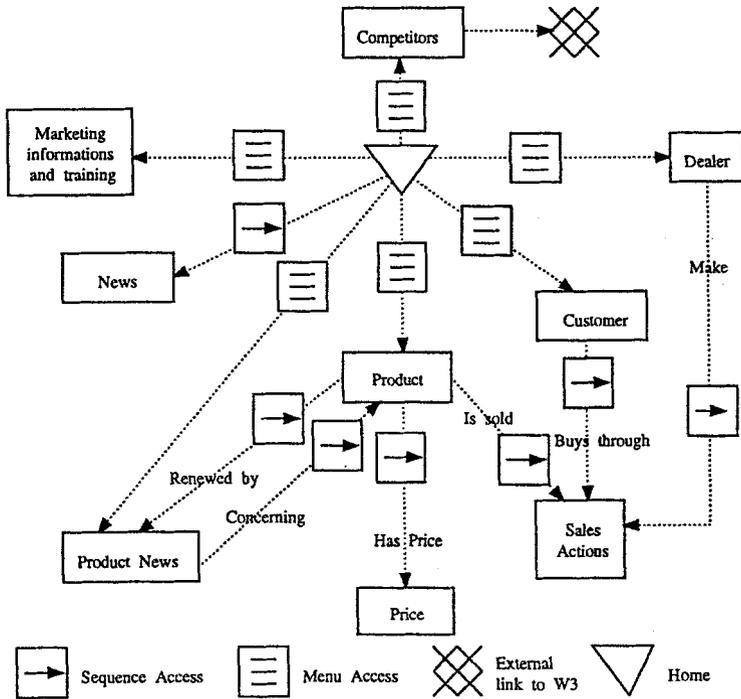


Fig. 1. RMDM schema of the sales department Intranet

related page. From the product news a user can access to the related product information through a sequential link.

### 3.3 Distribution Specification

This phase is required in order to choose the architecture of the distributed system and collect information about distribution, starting from the knowledge achieved from the requirement analysis. In particular, we need to observe the main fluxes of data from servers to users, that is retrieval and updates of the information. A possible fragmentation of entities is suggested and represented through a partitioning table.

The first problem that the designer has to solve concerns the network architecture, that is the number and the position of each server. We could place servers following some empirical considerations about corporation structure and location of users, derived from the segmentation tree. The sales Intranet will have three servers, one for each geographical area. ( $S_{eu}$ ,  $S_{am}$  and  $S_{aa}$ ).

The design of data distribution requires to evaluate entity fragmentation and allocation.

- *Fragmentation* is the process of division of entities into several units of allocation [5, 6]. Fragments are homogeneous collections of information with

a distinguishing property. For Database Systems, fragmentation can be either *horizontal* or *vertical*. In the first case, objects are divided in subsets of instances, and a projection of the object over a subset of his attributes is made otherwise. In the case of Web Systems, information can be fragmented with techniques similar to those adopted in distributed database design when the information is structured; when information is unstructured, it can still be horizontally partitioned, according to attributes associated to the text, or with derived partitioning from related entities. We can assume here that fragments are disjoint, that is that an instance can belong only to a fragment.

- *Allocation* is the process of mapping each fragment to one or more site. As in database distribution design [5, 6], we can distinguish between *redundant* and *non-redundant* allocation (a fragment is allocated to several sites in the first case, to one otherwise). A redundant allocation strategy is useful when the ratio between retrievals and updates is high, but with such a choice we have to face the problem of data consistency. We will see how the redundant allocation strategy can be applied usefully to the case of the web design.

In a distributed system we can distinguish between classes of different information, according to their local or global importance. We can have *locatable*, *non-locatable* and *partially locatable* entities:

- an entity  $E$  is *locatable* when it is possible to define a logical relation from the entity to the site set,  $\mathcal{R} : \mathcal{E} \rightarrow \mathcal{S}$ , that is between the information content and the specific site. In a locatable entity there exist an attribute  $A_k$  whose elements  $e_j$  can be associated to a site  $S_i$
- an entity  $E$  is *non-locatable* when its information content is general and not in relation with the site set. In this case, we can not define a relation as for locatable entities;
- an entity  $E$  is *partially locatable* when we can group instances into disjointed sub-entities, some of them locatable and some other non-locatable. The union of all sub-entities gives the starting entity. Note that in web applications entities are less structured than in conventional databases. Here different instances of an entity can have a different logical structure.

From the RMM schema of the sales Intranet (Fig. 1) one can observe that the entity *Customer* is locatable, because a customer has a precise geographical location. We can individuate a fragmentation of this entity according to the values of attributes address of customers. *Product* can be similarly fragmented according to locations, in a derived way, if it is the case the sales of certain products are specific to given geographical areas. Finally, the *News* entity can be separated between news of general and local interest. We are in the case of partially locatable entities.

In Table 1 entities are divided into classes.

The classification of entities into categories is the first step towards partitioning. We can after divide entities into subsets that can be possible units of

Table 1. Categories of entities

<i>Locatable</i>	<i>Non locatable</i>	<i>Partially locatable</i>
Dealer	Product News	News
Customer	Prices	Training
Sales	Competitors	Product

allocation. Locatable entities can be subdivided into horizontal subsets of instances; every instance of the fragment can be put into relation  $\mathcal{R}$  with the site. Note that fragmentation of an entity is independent from allocation, in fact an entity can be partitioned, but its fragments can be at the end allocated to a single server. The fragmentation process is made according to the logical structure, regardless from the use. It seems to be easier starting the fragmentation phase with locatable entities; we will obtain for each entity as many fragments as the sites. Partially locatable entities can be structured as locatable with the addition of a fragment, containing the instances belonging to the non-locatable sub-entity.

Table 2. Partitioning Table

Entity	Fragments	Strategy
<i>Dealers</i>	$D_{am}, D_{eu}, D_{aa}$	one fragment per site, by area belonging
<i>Customers</i>	$C_{am}, C_{eu}, C_{aa}$	one fragment per site, by area belonging
<i>Sales</i>	$Sl_{am}, Sl_{eu}, Sl_{aa}$	one fragment per site, by area where the sale action took place
<i>News</i>	$W_{am}, W_{eu}, W_{aa}, W_{gen}$	one fragment per site, by interest area and one fragment for general news
<i>Training Marketing</i>	$T_{am}, T_{eu}, T_{aa}, T_{gen}$	one fragment per site for specific information, one for general
<i>Competitors</i>	$Co$	non-locatable and not fragmented
<i>Products</i>	$P_1, P_2, P_3, P_4, P_5, P_6, P_7$	derived partitioning, based on the relationship <i>is sold</i> <sup>a</sup>
<i>Product News</i>	$PN_1, PN_2, PN_3, PN_4, PN_5, PN_6, PN_7$	derived partitioning, based on the relationship <i>concerning</i> <sup>b</sup>
<i>Prices</i>	$Pc_1, Pc_2, Pc_3, Pc_4, Pc_5, Pc_6, Pc_7$	derived partitioning, based on the relationship <i>has price</i> <sup>b</sup>

<sup>a</sup> For each product, area of sale deducted from the relationship with *Sales*. Fragments derived:  $P_1$  (products sold mainly in Europe),  $P_2$  (in America),  $P_3$  (in Australasia),  $P_4$  (in Europe and America),  $P_5$  (in Europe and Australasia),  $P_6$  (in America and Australasia) and  $P_7$  (products sold in all areas).

<sup>b</sup> Fragments obtained following the structure of *Products* fragmentation.

The fragmentation of non-locatable information can be achieved through a derived partitioning [5] strategy. When a non locatable entity  $E_2$  is in relation

(in the RMDM model) with  $E_1$ , already partitioned in fragments  $F_{1i}$ , we can subdivide  $E_2$  in fragments  $F_{2i}$ , containing the set of instances related to those of  $F_{1i}$ . We constrain fragments to be disjoint. At the end of the partitioning phase, a partitioning table (Table 2) reports the list of fragments and the description of the associated fragmentation strategy. The partition of *Customers*, *Dealers* and *Sales* in geographical zones is natural, because this belonging is easily determined. The fragmentation of *Products* entity is derived. We divide products according to the area where they are mostly sold.

Once the information has been partitioned, one has to analyze how each fragment is accessed from users and updated from providers. In particular, the retrieval frequency with direction and type of access has to be deeply understood, as well as the frequency and the volume of an updating operation. This analysis is preliminar to the allocation phase, because the placement strategy is based on an interpretation of these data.

For each entity, we have to build a *retrieval table*, containing the frequency of access  $fr$  (expressed in number of accesses to pages containing the entity information), and, for each fragment, the probability that the retrieval comes from each site. These data are obviously estimate of real frequencies and probability, and have to be collected through an on-the-field research, analyzing every user's need.

The Retrieval Tables of entities *Dealers* and *Product* are reported as example in Tables 3 and 4. The following data are completely fictional, and they are based on an approximate evaluation of how data can be accessed.

Table 3. retrieval table for *Dealers*

<b>Dealers</b> $fr: 50$ acc. per day			
<i>retrieval prob.</i>	$S_{eu}$	$S_{am}$	$S_{aa}$
$D_{eu}$	80	10	10
$D_{am}$	10	80	10
$D_{aa}$	10	10	80

Table 4. retrieval table for *Products*

<b>Products</b> $fr: 1000$ acc. per day						
<i>retrieval prob.</i>	$S_{eu}$	$S_{am}$	$S_{aa}$	$S_{eu}$	$S_{am}$	$S_{aa}$
$P_1$	80	10	10	$P_5$	45	10 45
$P_2$	10	80	10	$P_6$	10	45 45
$P_3$	10	80	10	$P_7$	33	33 33
$P_4$	45	45	10			

Note that *Dealers* is a locatable entity and *Products* is partially-locatable. The problem of updating an entity affects remarkably the allocation of its fragments. In the *Updating Table* (Table 5) we describe updates characteristics for each entity. In the table, we assume that all fragments of a given entity have identical update characteristics, hence these properties can be referred to the whole entity and can be assumed valid for each of its fragments. In the table, we report the frequency of updates in terms of mean time elapsed between two updates, the mean volume of transferred data for a single operation, expressed in pages, and if the refresh is made locally on the server or by a *global* webmaster from the main server. This information is necessary for the cost evaluation of an update action.

Table 5. Updating table

	Refresh	Freq.	Volume		Refresh	Freq.	Volume
<i>D</i>	local	monthly	50 pgs.	<i>T</i>	global	monthly	10 pgs.
<i>C</i>	local	weekly	100 pgs.	<i>Co</i>	global	seldom	5 pgs.
<i>S</i>	local	every hour	1 pg.	<i>P</i>	global	monthly	100 pgs.
<i>W</i>	global	every hour	1 pg.	<i>PN</i>	global	daily	10 pgs.
<i>Pc</i>	global	monthly	100 pgs.				

### 3.4 Distribution Logical Design

This phase concerns fragment allocation, that is the assignment of all fragments to one or more sites. The allocation can be *non-redundant* when a fragment is mapped to exactly one site, or *redundant* when each fragment is allocated to two or more sites. For a given fragment  $F_i$ , the first characteristics that we have to analyze for the decision towards redundant or non-redundant allocation strategy is the distribution of frequency accesses.

When the retrieval probability  $p_{ij}$  for a server  $S_j$  is greater than a threshold value  $T$ :

$$\exists S_j \mid p_{ij} > T \quad (1)$$

the retrieval distribution is lumped, that is the fragment is retrieved mainly from just one site. Hence, the fragment can be allocated only in  $S_j$ , that is we have a non redundant allocation of  $F_i$ . On the other hand, if no server satisfies (1), the evaluation of the choice towards a redundant allocation of the fragment is suggested.

The distribution of the information is recommended when the ratio between retrieval and updates is high, since updates from all sites have to be considered to assure data consistency.

The choice between centralisation or distribution of an entity fragment can be supported by the evaluation of the *network load*, that is the intensity of use

of network resources. We can define two functions that analytically reproduce the cost of the network use with respect to a fragment, both when data on the main server are retrieved from local users ( $CR_i$ ) and when information resident on local servers is updated from the central webmaster ( $CU_i$ ).

$$CR_i \triangleq \sum_{j=1}^N (\alpha \cdot fr_i \cdot p_{ij}) \quad (2)$$

$$CU_i \triangleq N \cdot fu_i \cdot V_i \quad (3)$$

In equations (2) and (3)  $fr_i$  and  $fu_i$  are respectively the retrieval and updating frequencies of the entity to which the  $i$ -th fragment belongs,  $p_{ij}$  is the probability that the  $i$ -th fragment is retrieved from site  $j$ ,  $V_i$  is the mean volume per updating operation related to the entity of the  $i$ -th fragment (see Tables 3 and 4),  $N$  is the total number of sites and  $\alpha$  is the cost of the user connection. This choice is made because, for instance, a connection via modem used by a salesman is more expensive than a permanent connection, so we need to quantify this important aspect.

The knowledge of  $CR_i$  and  $CU_i$  allow us to decide the best choice between a distributed or non-distributed allocation for a fragment. When  $CR_i$  is greater than  $CU_i$ , retrieving the fragment from the main server is more expensive than making it locally available, so distribution is recommended. When the updating on the local server is heavy, and the cost of retrieving the non-distributed information is lower, the distribution of the fragment is otherwise not recommended.

The allocation phase consists in a decision process, represented by the decision tree of Fig. 2, ending with the allocation of each fragment to one or more servers.

Fragments are considered one by one, and submitted to the decision process. From the evaluation of retrieval probabilities and their matching with the threshold  $T$ , combined with the evaluation of  $CR_i$  and  $CU_i$ , we can allocate a fragment to a server according to the following main principles:

- for information retrieved mainly from a single site redundant allocation is useless. When the information access comes otherwise from several servers, redundant allocation is a possibility to evaluate;
- when an entity does not need frequent refresh we can distribute it. In fact, we do not have the problem of data inconsistency, and the network load due to updating is minimal;
- when fragments of locatable entities are locally updated, they can be allocated to the corresponding local site;
- in the case fragments are updated from a central webmaster (globally updated), the comparison between  $CR_i$  and  $CU_i$  allows us to decide for distribution. As seen above,  $CR_i > CU_i$  means that the fragment has to be placed in one or more remote servers, otherwise we allocate it to the main one.

In the example of the sales Intranet, we can allocate fragments of each entity, following the decision tree steps. *Dealers*, *Customers* and *Sales* entities retrieval

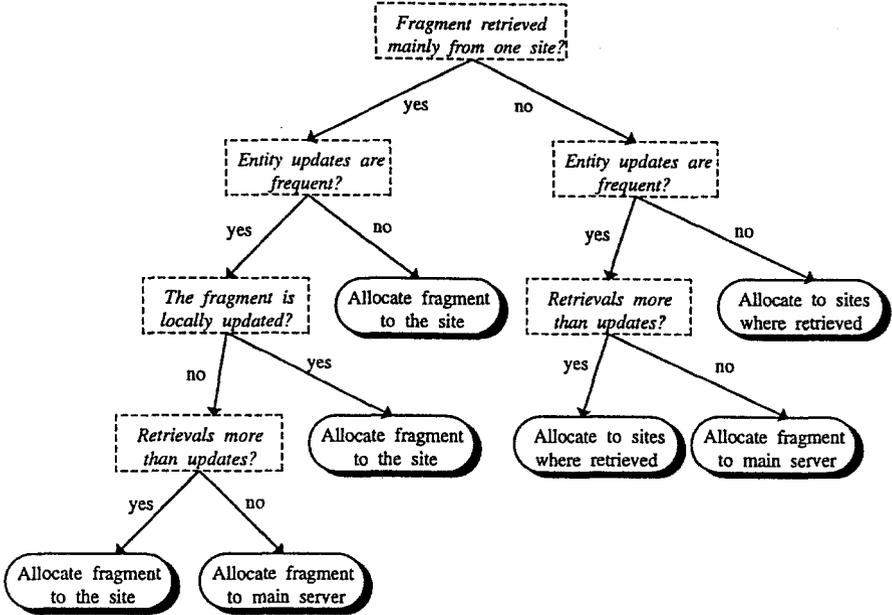


Fig. 2. The decision tree of the allocation process

tables show that fragments are likely to be retrieved from a specific local site. *Dealers* and *Customers* do not need frequent updates, and *Sales* is locally updated. We decided to allocate each fragment to a single local server. *Products* and *Prices* fragments do not need frequent updates as well. About *Prices*, we decide to allocate  $P_1$ ,  $P_2$  and  $P_3$  to a single local server,  $P_4$ ,  $P_5$  and  $P_6$  with redundancy to two servers and  $P_7$  to all the three servers. *News* and *Product news* need a constant refresh, so they are placed on the main server, allowing remote access from all sites. On the other hand *Training* is seldom updated:  $T_{eu}$ ,  $T_{am}$  and  $T_{aa}$  are allocated to single sites, and  $T_{gen}$  with redundancy to all servers. After the allocation phase, we are able to build an *Allocation table* Table 6, reporting where fragments are allocated.

Table 6. Allocation table

Server	Fragments
$S_{eu}$ (main)	$D_{eu}, C_{eu}, Sl_{eu}, T_{eu}, T_{gen}, Co, W, PN, P_{1,4,5,7}, PC_{1,4,5,7}$
$S_{am}$	$D_{am}, C_{am}, Sl_{am}, T_{am}, T_{gen}, P_{2,4,6,7}, PC_{2,4,6,7}$
$S_{aa}$	$D_{aa}, C_{aa}, Sl_{aa}, T_{aa}, T_{gen}, P_{3,5,6,7}, PC_{3,5,6,7}$

For simplicity, in Table 6 we denote with  $W$  and  $PN$  all fragments of *News* and *Product News* entities.

### 3.5 Local Design

This step concerns the development of local RMDM schemas, one for site, derived from the global schema and from the allocation table. In the local RMDM sub-schema we report the information structure of the site, contained in a dashed box. Entity fragments are represented by continuous line boxes, and arrows show relational links. When the link connects local fragments the relation is internal. Links can be, on the other hand, directed towards fragments belonging to the main site, represented in dotted boxes placed outside the dashed border of the local site. In addition, we can have menu links that allow the choice among local or external fragments. These particular kind of link are represented as internal ones, but they are marked with a special symbol, reported in (Fig. 3).

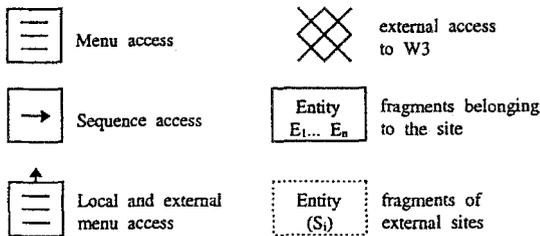


Fig. 3. Symbols of RMDM extended to distributed environments

In local schema the menu access and the sequence access symbols maintain their meaning, and they can be used for internal as well as external links. In Figs. 4 and 5 we report, as example, local schemas for the european and the american site, referred to the sales Intranet example.

Local schemas based on the extended RMDM model clearly describe the structure of each server: the internal links between entities are reported, as well as the relations with information available on remote servers. We can think that such an organisation of the Intranet system allow us to minimize the network load, distributing or concentrating the information according to the modality and frequency of retrieval.

The last phase of the design process (the *physical design*) concerns the determination of a lower level representation in order to develop the final HTML code of single pages. This representation of the system has to include the internal structure of entities and the description of links between instances. The description of this last phase is outside of our scope. For more details about low level design see, as example, the HDM approach [9] or the slicing approach [13] in RMM methodology.

## 4 Concluding Remarks

The problem of distributing information on the Web is becoming of extreme importance, for the fact that we are assisting to a sharp growth of Intranet

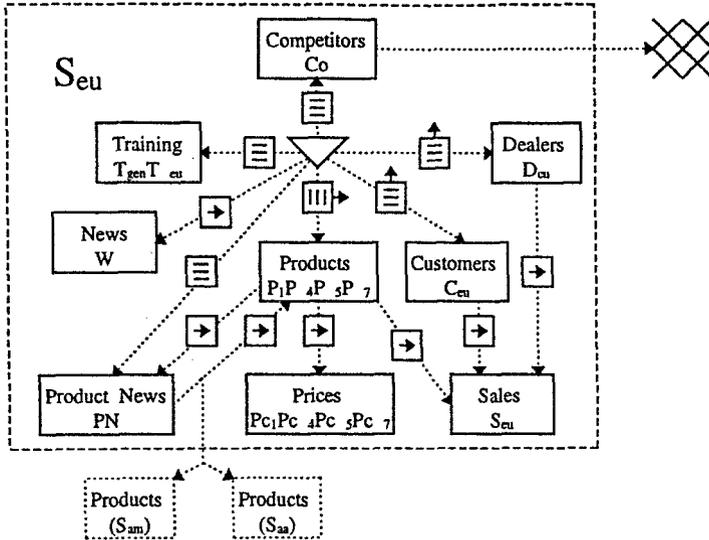


Fig. 4. Extended RMDM local schema - European site

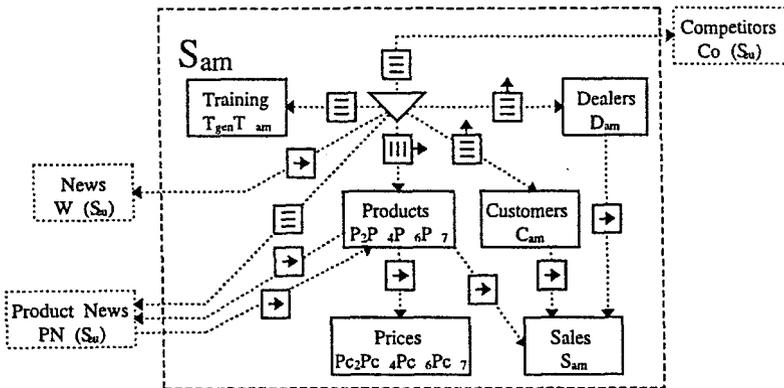


Fig. 5. Extended RMDM local schema - American site

solutions for internal communication in big companies.

A systematic methodology, based on a top-down approach, has been presented in this work, with the aim of giving the guidelines for web design and optimising the network load. The design of the global structure, combined with the specification of data and user distribution, leads to a decision process where the allocation of fragmented entities take place. For each fragment, the opportunity of a redundant allocation is evaluated, in alternative with data centralisation or with non-redundant data placement.

The creation of local schemas can be considered the heart of the developed methodology. The schemas are based on an extension of RMDM model, and report the information together with the access structure. Through these models,

the complex structure of a distributed web system is clearly represented.

The example of the sales department of an international company has been used to illustrate the different phases of the Web System development process.

## References

1. V. Balasubramanian, B. Min Ma, J. Yoo. A Systematic Approach to Designing a WWW Application. *Communications of the ACM*, Vol. 38, No. 8, 47-48, August 1995.
2. R. Bernard. The Corporate Intranet. Wiley Computer Publishing, New York, NY, USA, 1996.
3. T. Berners-Lee, R. Caillau, A. Luotonen, H. Frystyk Nielsen and A. Secret. The World Wide Web. *Communications of the ACM*, Vol. 37, No. 8, 76-82, August 1994.
4. F. Casati, B. Pernici. A Methodology for the Design of WWW sites and its Application to Distance Education. *Workshop on Web-Based Education, Intl. WWW Conference*, Paris, May 1996.
5. S. Ceri, B. Pernici, G. Wiederhold. Distributed Database Design Methodologies. *Proceedings of the IEEE*, Vol. 75, No. 5, 533-546, May 1987.
6. S. Ceri, B. Pernici, G. Wiederhold. Optimization Problems and Solution Methods in the Design of Data Distribution. *Information Systems*, Vol. 14, No. 3, 261-272, March 1989.
7. G.P. Cline. Web servers and the rise of the corporate Intranet. *International Journal of Network Management*, Vol. 5, No. 6, 322-325, Wiley & Sons Ltd, Chichester Engl., Nov-Dec 1995.
8. R. Eros. Learning to love the Intranet. *Telecommunications (International Edition)* Vol. 30, No. 6, Horizon, House Norwood, MA, USA, June 1996.
9. F. Garzotto, P. Paolini, D. Schwabe. HDM-A Model-Based Approach to Hypertext Application Design. *ACM Transactions on Information Systems*, Vol. 11, No. 1, 1-26, January 1993.
10. C. Ghezzi, M. Jazayeri, D. Mandrioli. Fundamentals of Software Engineering. Englewood Cliffs: Prentice-Hall, 1991.
11. C. Hildebrand. Surf & Turf. *WebMaster*. July 1996.
12. T. Horgan. The Design and Implementation of a Corporate Web. *Seminar on WebMaster Magazine*, <http://www.cio.com/WebMaster/sem1.intro.html>, 1996.
13. T. Isakowitz, E. A. Stohr, P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, Vol 38, No.8, 35-44, August 1995.
14. H. Schulzrinne. World Wide Web: Whence, Whither, What Next? *IEEE Network*, 10-17, March/April 1996.
15. I. Sommerville. Software Engineering. Addison-Wesley, 1989.