Modelling Static and Dynamic Aspects of Hypermedia Systems

Paloma Díaz¹, Ignacio Aedo¹, Fivos Panetsos²

¹ Laboratorio DEI. Computer Science Department. Universidad Carlos III de Madrid. C/ Butarque 15, 28911 Leganés. Spain.

Abstract

The static and dynamic properties of the Labyrinth model, used for the definition and management of hypermedia applications, are presented in this paper. The most relevant characteristics of Labyrinth are the following: the structure of applications can be modelled by means of composite objects; the reactive nature and temporal dimension of multimedia contents are considered; and security policies to preserve information confidentiality and integrity can be defined.

1. Introduction

Hypermedia systems differ from other software applications in critical issues, such as the inclusion of reactive and dynamic information and their navigational behaviour. Due to these special characteristics, the use of traditional models and methodologies is not suitable for the development of hypermedia systems [1]. However, the majority of hypermedia-oriented models is very restrictive, since they depend on physical data representations or platforms. Even though general and abstract models do exist (such as [2, 3, 4]), some relevant issues still remain not addressed, such as the definition of composite objects with heritage mechanisms and security policies that preserve information confidentiality and integrity. In this context, we have developed Labyrinth, a model for the static and dynamic definition of hypermedia applications.

2. The Labyrinth model

Labyrinth provides elements to develop a static definition of hypermedia applications as well as operations to manage their dynamic usage.

2.1 Static definition of applications

In Labyrinth, hypermedia applications are defined as the union of a "Basic Hyperdocument" and a series of "Personalized Hyperdocuments". The "Basic Hyperdocument" formalizes a common hyperdocument by means of seven sets of elements (Users, Nodes, Contents, Anchors, Links, Attributes and Events) and four functions (location, attributes list, events list and access list). A "Personalized Hyperdocument" belongs to a particular user, either individual or group, and it is defined by means of six sets of elements (Nodes, Contents, Anchors, Links, Attributes, Events) and three functions (location, attributes list and events list) which contain both components of the Basic Hyperdocument that have been adapted or new ones. These elements are described below.

Users - \( U = \{ U_i \mid i = 0, \ldots, n, n \in \mathbb{N} \} \), where

\[ U_i = (Userld_i, UserType_i, UsersList_i) \]

The set of users \( U \) contains both individuals and groups. The inclusion of "U" in the model allows to design a users' structure, to formally specify a security policy and to include personalizations. Each user \( U_i \) is defined as an identifier (Userld_i) and a type (UserType_i) which states whether it is an individual or a group. If the user is an individual, a list of groups is added to its definition (UsersList_i). If the user is a group, a list of members is added to its definition (UsersList_i).

Nodes - \( N = \{ N_i \mid i = 0, \ldots, n, n \in \mathbb{N} \} \), where

\[ N_i = (NodeId_i, NodeCategory_i) \]

Labyrinth separates the abstract containers (nodes) from the information items (content), in order to provide a way to include the same content into different nodes without duplicating information. Each node \( N_i \) consists of an identifier (NodeId_i) and a security category (NodeCategory_i) which establishes the kind of operations that can be performed on the node and its components (links, attributes and events). Security category can be set to "browsing", "personalizing" or "editing". A "browsing" node can only be visited; a "personalizing" node can also be adapted to the users' needs and preferences; and, finally, an "editing" node can also be modified in the "Basic Hyperdocument" to spread updates through the users of the application. Nodes can be composite objects that, combined with typed links, represent aggregations and generalizations.

Contents - \( C = \{ C_i \mid i = 0, \ldots, n, n \in \mathbb{N} \} \), where

\[ C_i = (ContentId_i, ContentCategory_i, ContentType_i) \]

A content \( C_i \) is an information piece that consists of an identifier (ContentId_i), a security category (ContentCategory_i), used as in the nodes, and a type (ContentType_i), which defines the nature of the content.
The access list function, "ac", is used to preserve information confidentiality and integrity. To ensure confidentiality, the "ac" function of contents and nodes can be set to 0 for those users who can not even retrieve them.

3. Conclusions

The most relevant features of the Labyrinth model are:
- Its elements do not depend on physical constraints or authoring tools, allowing the design of platform-independent hypermedia applications.
- Unstructured heterogeneous information can be categorized in multiple and interconnected levels by means of composite nodes and typified links.
- Hypermedia elements can be defined and managed.
- The reactive nature of multimedia information can be gathered by using events.
- Anchors can be defined into any spatial and temporal circumstances. Moreover, events allow conditional links as well as links to external applications to be represented.
- Problems arising in collaborative environments, like information sharing, personalization, version control and information security, are addressed in the model.

4. References