IIPS: an Intelligent Information Presentation System

Yuangui Lei, Enrico Motta and John Domingue
Knowledge Media Institute
The Open University
Walton Hall, MK7 6AA
{y.lei, e.motta, j.b.domingue}@open.ac.uk

Abstract

This paper presents the framework of an Intelligent Information Presentation System (IIPS), which provides intelligent interface presentation support for data-intensive web-based applications through the use of ontologies to drive the web site generation and maintenance process. IIPS defines a comprehensive set of ontologies to model the navigational structure, the compositional structure, and the user interfaces of data-intensive web sites, and provides a suit of tools to support site generation, maintenance, and personalization.

Keywords

Model-based, Intelligent Interface Presentation, Web Site Management System

INTRODUCTION

Web application development and maintenance are becoming more and more complex, costly and time-consuming. To address this problem, model-driven methodology has been used to try to simplify and eventually automate the whole process of generation and maintenance of data-intensive web applications, by modeling domain applications and data-intensive web sites. Many efforts have been made to achieve this goal [1,2,3]. However none of them can provide efficient support for the whole process of generation and maintenance of data-intensive web applications. First, they mainly focus on expression of data content, and fail to model interface presentation issues, such as page layouts and the graphic user interfaces. Second, they either do not address the problem of site maintenance, or simply support site maintenance only at content level.

IIPS overcomes the problems through defining a set of comprehensive site ontologies to model the navigational structure, the compositional structure, and the user interfaces of data-intensive web sites, and providing a suit of tools to support site generation, maintenance, and personalization.

THE FRAMEWORK OF IIPS

Figure 1 shows the framework of IIPS. It accepts a domain ontology as input, and produces a customizable and adaptive web site.

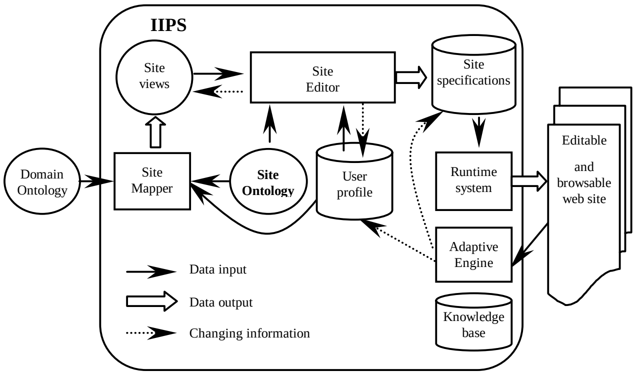

Figure 1 IIPS Framework

The *Domain Ontology* serves as an underlying structure to describe data content in the target web site. It describes concepts and relationships in the specified domain. Concepts and their hierarchy structure can drive the design of a site view, and the relationships are very useful to drive efficient navigation and information retrieval.

The *Site Ontology* describes the navigational and compositional structure, and interfaces of a web site. IIPS extends pre-existing approaches to site modeling as it models not only traditional database-driven web pages, but also web sites. It defines four top classes to model a general web site: site, resource, component, and layout, and makes use of *an interface ontology* to describe the main graphical entities used in web sites.

- The <u>*site class*</u> models a web site to be a logical collection of resources. It has an URI (Uniform Resource Identifiers) to distinguish it from other sites, various resources to achieve the site goal, and an index resource to help users to walk through the web site.
- The resource class defines anything that has a URI to identify it as [4]. It may be a part of web page, a web page, or even an entire web site. It has sub-resources, components, and layouts. The most important, it must have a URI to distinguish it.
- The component class models contents to compose resources. It has two subclasses: input-component and

output-component. The input-component class models components which allow users to input information. It contains input-widgets which allow users to enter information, and data which connects input-widgets with domain data. The output-component class describes components to present information. It is made up of output-widgets which present information, and data which tells what to display. IIPS defines several component primitives to support the site modelling. The search-component allows users to make a search. The ka-component enables users to input facts into a knowledge base. The index-component lists the index of a set of instances according to their ordering keyword, with links and but without presenting the detailed information for each instance. The data-component presents detailed information about one instance of a domain concept.

- The layout class defines the layout style for resources and components to display contents.
- The interface ontology defines user interface concepts and design guidelines including container prototypes, widget prototypes, and data-widgets mapping rules. It defines a *container class* to model prototypes to hold resources and a *widget class* to describe widget prototypes. The *interface ontology* performs as an interface mapping knowledge base for the Site Mapper. It is integrated into the site ontology through the component class definition in the site ontology, which specifies data content and graphic user interfaces.

The *Site Mapper* generates default site views by mapping the site ontology to a domain ontology for predefined user groups. IIPS predefines four user groups: developers, web masters, advanced users and general users. Each user group has a default site view. Default site views contain resources restricted to the user group. It serves as basis for developers to edit site views for user groups or users, and for users to customize site views.

The *Site Editor* supports developers and web masters editing site views, modifying page contents, creating new web pages, and extending the site ontology. End users can also utilize it to customize a web site to their individual need according to his or her access right defined by web masters.

The *Runtime System* accepts the declarative site specifications as input and dynamically generates web pages. It is responsible for checking the login information, finding the declarative specifications consist with the user name or the user group he or she belongs to, generating web pages, establishing connection between web pages and domain knowledge base, executing data transactions, and invoking the Adaptive Engine to observe user's interactions and present an adaptive web site to individual users.

The *Adaptive Engine* aims to provide adaptive site views and interfaces to individual users. IIPS employs the overlay user modeling approach to modeling a user's interest level about each domain concepts by establishing an overlay user model to record the visiting and operating frequency of a user to each domain concept. In addition, IIPS uses an approach based on stereotypes to classify the user groups and to associate rights of accessing the target web site with these user groups.

CONCLUSION AND FUTURE WORK

IIPS is an intelligent information presentation system which aims to provide intelligent interface presentation support for data-intensive web-based applications through the use of ontologies to drive the web site generation and maintenance process. To achieve this goal, a comprehensive site ontology and interface ontology are defined to model the navigational structure, the compositional structure, and user interfaces of data-intensive web sites, and a suit of tools is provided to support site mapping, site editing, site specifications generating, web pages generating, site customization, and site personalization.

Compared to other web site management systems, IIPS framework has three main strengths. First, unlike most other data-intensive web site management systems, which mainly focus on the expression of data content on web pages, IIPS models interface presentations thoroughly through modelling layouts and graphic user interfaces. Second, it supports site maintenance both at content level and at the site specification level. In particular, it provides both knowledge acquisition forms to allow advanced users to make contributions to individual pages, and a Site Editor, which allows users to customize the site views through manipulating the domain ontology and the site ontology. Finally, it provides comprehensive adaptive facilities to adapt and personalize a web site for a particular user.

Future work involves designing explicit detailed mapping rules to map domain ontology with the site ontology and the interface to generate site specifications, and developing the IIPS tool suite.

REFERENCES

[1] Stefano Ceri, Piero Fratenali, Aldo Bongio, Web Modeling Language (WebML): a modeling language for designing Web sites, WWW9 Conference, Amsterdam, May 2000.

[2] Mary F. Fernandez, Daniela Florescu, Alon Y. Levy, and Dan Suciu, Declarative Specification of Web Sites with Strudel, VLDB Journal 9(1):38-55,2000.

[3] T. Isakowitz, E.A. Stohr and P. Balasubramaninan, RMM: A Methodology for Structured Hypermedia Design, Communications of the ACM, August 1995.

[4] Resource Description Framework (RDF) Model and Syntax Specification, available at <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.